

Министерство образования и науки РФ  
Федеральное государственное бюджетное образовательное  
учреждение высшего образования  
«Уральский государственный педагогический университет»  
Институт математики, информатики и информационных технологий  
Кафедра информационно-коммуникационных технологий в образовании

# **РАЗРАБОТКА И СОЗДАНИЕ САЙТА НА ОСНОВЕ CMS**

*Выпускная квалификационная работа  
бакалавра по направлению подготовки  
230400 – Информационные системы и технологии*

Исполнитель: студент группы БС-41  
Института математики, информатики и ИТ  
Коханова П.С.

Руководитель: кандидат филологических наук,  
доцент кафедры ИКТО  
Грушевская В.Ю.

Работа допущена к защите  
«12» мая 2016 г.  
Зав. кафедрой \_\_\_\_\_

Екатеринбург – 2016

## **Реферат**

Коханова П.С. РАЗРАБОТКА И СОЗДАНИЕ САЙТА НА ОСНОВЕ CMS, выпускная квалификационная работа; 69 стр., рис 56, табл. 1, библи. 30 назв..

*Ключевые слова:* РАЗРАБОТКА САЙТА, CMS.

*Объект разработки* – организации детских праздников «Таити».

*Цель работы* – разработка и создание веб-сайта.

В работе описаны результаты проектирования разработки сайта и создания сайта.

Сайт реализован с использованием CMS WordPress.

## Оглавление

<b>ВВЕДЕНИЕ.....</b>	<b>4</b>
<b>ГЛАВА 1. СИСТЕМА УПРАВЛЕНИЯ КОНТЕНТОМ И ИХ ИСПОЛЬЗОВАНИЕ В СОВРЕМЕННЫХ УСЛОВИЯХ.....</b>	<b>6</b>
1.1 ТЕХНОЛОГИЯ СОЗДАНИЯ ВЕБ-САЙТА .....	6
1.2 СРАВНИТЕЛЬНЫЙ АНАЛИЗ CMS.....	8
1.2.1 Анализ WordPress.....	10
1.2.2 Анализ MODx .....	13
1.2.3 Анализ Joomla.....	17
1.2.4 Результат и вывод рассматриваемых CMS.....	20
1.3 ФОРМАЛИЗОВАННОЕ ОПИСАНИЕ ТЕХНИЧЕСКОГО ЗАДАНИЯ.....	20
<b>ГЛАВА 2. ОПИСАНИЕ ПРОЦЕССА РАЗРАБОТКИ ВЕБ-САЙТА .....</b>	<b>24</b>
2.1 МОДЕЛЬНЫЕ ПРЕДСТАВЛЕНИЯ ОБЪЕКТА РАЗРАБОТКИ.....	24
2.1.1 Этапы проектирование веб-сайта.....	24
2.1.2 Функциональная модель разработки веб-сайта .....	27
2.1.3 Структура веб-сайта.....	29
2.2 ОПИСАНИЕ ПРОДУКТА (РЕЗУЛЬТАТА РАЗРАБОТКИ).....	31
2.2.1 Постановка целей и задач .....	32
2.2.2 Реализация веб-сайта .....	33
2.2.3 Тестирование веб-сайта.....	50
2.2.4 Безопасность веб-сайта.....	53
2.3 РЕКОМЕНДАЦИИ ПО ИСПОЛЬЗОВАНИЮ И НАСТРОЙКЕ ВЕБ-САЙТА НА CMS WORDPRESS	55
<b>ЗАКЛЮЧЕНИЕ .....</b>	<b>66</b>
<b>СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ.....</b>	<b>67</b>

## **Введение**

С появлением веб-технологий жизнь людей полностью изменилась. Большой объем информации находится на виртуальном пространстве. По данным ФОМ, осенью 2014 года 72,3 миллиона россиян (62% совершеннолетнего населения страны) пользовались Интернетом хотя бы раз в месяц, а почти 60 миллионов человек выходили в сеть ежедневно. [1] Всемирная паутина пестрит всевозможными веб-сайтами с различной информацией: развлекательные порталы, интернет-магазины, услуги и т.д. Множество организаций и крупных компаний стали использовать информационные технологии в свою пользу.

На сегодняшний день веб-сайт считается не только информационным инструментом, но и полноценной рекламной площадкой, притягивающей новых покупателей. Существует колоссальное число информационных сайтов, предоставляющих полные сведения по указанным запросам пользователя. Если раньше информацию стоило искать по печатному изданию, то сейчас достаточно выйти в Интернет.

Одним из основных инструментов разработки и создания сайта является системы управления контентом (CMS). CMS отводится существенная доля в формировании Интернета. По сути, с появлением CMS сняты технические лимиты в создании собственного сайта – достаточно только понять интерфейс, разработать структуру и получить готовый сервис. Все это стало возможным благодаря широкому распространению системы управления контентом, большая часть из которых не требует специальных знаний в сайтостроении.

Актуальность исследуемой нами проблемы обоснована тем, что в наши дни сайт, сформированный на основе такого рода концепции разработки, в большинстве ситуаций преобразуется в практичный бизнес-механизм, которым смогут воспользоваться непосредственно работники организации. С помощью CMS структура веб-сайтов видна и понятна не только разработчикам, но и заказчикам проектов, что позволяет им понимать механизм работы сайта.

Раньше при изменении дизайна сайта или обновлении новостей, приходилось передавать информацию веб-мастеру, который в корне менял все, при глобальных обновлениях приходилось с нуля писать сайт. CMS позволяет своевременно заполнить, обновить информацию и разместить ее на сайте. Также пропадает необходимость в дополнительных специалистах (веб-мастер, системные администраторы, программисты и прочие), что значительно уменьшает расходы организации.

Отталкиваясь от вышесказанного выбранная нами, проблема считается современной и важной.

Целью выпускной классификационной работы является разработка и создание веб-сайта. Разработанный сайт имеет возможности:

- публиковать в Интернете об услугах, акциях и новостях организации;
- продемонстрировать мультимедийную и графическую информацию;
- установить обратную связь с клиентом;

Объектом исследования является организации детских праздников «Таити».

Для реализации поставленной цели были выявлены следующие задачи:

- сравнительный анализ популярных CMS;
- разработка дизайн-макета веб-сайта;
- верстка макета веб-сайта по разработанному и утвержденному дизайн-макету;
- создание основных разделов веб-сайта;
- обучение сотрудников Заказчика редактированию сайта.

# **Глава 1. Система управления контентом и их использование в современных условиях**

## **1.1 Технология создания веб-сайта**

На данный момент фактически любая организация имеет свой веб-сайт. Организация, идущая в ногу со временем, применяет информационные технологии – ведь это важный инструмент, позволяющий расширить поле маркетинговой работы и заинтересовать дополнительных посетителей.

С самого начала все сайты были статическими – html-страницы были связаны друг с другом при помощи гиперссылок. Статический веб-сайт – это сайт, состоящий из статических страниц, которые располагаются на веб-сервере так, как видит его пользователь в браузере. Иными словами, документ лежит на сервере в собранном виде и загружается с сервера одним файлом с присоединенным к нему CSS стилям или скриптами. В основном такие сайты разрабатывались для представления общей информации об организации – сайты-визитки, состоящие из одной или несколько страниц. Любая страница помимо содержательной информации включает в себя – шапку сайта, навигацию, меню и др. Вследствие этого, на страницах отображаются конкретные документы, контент и дизайн, кроме того, при публикации новых страниц или редактировании уже имеющихся страниц надо учитывать каждую стилевую особенность и задавать стандартное оформление. Так же в ряд недостатков статических сайтов входит отсутствие готовых стандартных функций, например:

- Отсутствие регистрации и аутентификации;
- Отсутствие персонализации;
- Отсутствие чатов, форумов, гостевых книг;
- Отсутствие готовой интеграции с электронной почтой;
- Сложность в обновлении информации;

Сложно структурированные сайты, основанные на статических страницах, невыгодны и неудобны, ведь они несут в себе немалые финансовые потери (приходится нанимать веб-программистов), так же времени на манипуляции с

сайтом будет затрачено намного больше. Функциональность и дизайн фактически не разделяются, поэтому при изменении дизайна требуется постоянное сотрудничество дизайнера и программиста. Изменение структуры или навигации могут привести к возможности изменения структуры данных и к практически полному перепрограммированию самого сайта.

Стоит отметить, что сайты бывают еще и динамическими. Информация динамических страниц хранится не в виде готовых html-страниц, а в базе данных. Если в примере со статическим сайтом, при запросе пользователя в окне браузера отображается страничка в таком виде, в каком она существует, то при работе с динамическим сайтом страница, которую вы видите, не существует. Динамическая страница сделана из отдельных частей, которые веб-сервер собирает в целое. Когда сервер получает запрос на какую-либо информацию, он собирает страницу из нужных частей. Таким образом, может быть создан каркас сложного веб-сайта. При разработке динамического сайта требуется опытный программист, усложняется процесс постановки задач и увеличивается цена ошибок.

Отделение дизайна и содержимого сайта считается основной характеристикой динамических сайтов.

Но мы сталкиваемся со следующей проблемой – статические сложны в обновлении, редактировании и не подходят для сложно структурированных сайтов, поддержка динамического сайта требуют веб-программиста, что введет к увеличению расходов организации.

CMS предлагает компромисс между двумя подходами в сайтостроении. Потребность систем управления контентом начала проявляться при быстром увеличении роста информации на сайтах и так же быстро меняющиеся условия бизнеса. Так же необходимость на сайте различных автоматизированных механизмов привела к тому, что внедрение отдельных скриптов, написанных различными профессионалами, прекратили удовлетворять требованиям защищенности. CMS стандартизировал программные решения.

CMS обладает массой возможностей в создании веб-ресурса: с легкостью можно добавлять информацию, размещать галереи, ограничить доступ к информации, создать обратную связь и это всего лишь маленький список того, что можно сделать с помощью CMS.

## **1.2 Сравнительный анализ CMS**

Введя в поисковой строке «CMS», Вы обнаружите несколько десятков различных систем управления контентом. Каждая CMS отличается функционалом, дизайном и модулями. Фактически, на рынке представлены более тысячи CMS. Поэтому выбор наиболее подходящего решения, которое будет отвечать поставленным задачам, достаточно сложен. Перед тем, как выбрать CMS, нужно поставить перед собой следующий вопрос: каковы цели и требования от сайта? Например, если это будет интернет-магазин, то существуют отдельные CMS, которые специально разрабатывались только для интернет-магазина, то есть для создания блога такая система управления уже не подходит.

Для сравнительного анализа CMS необходимо сократить спектр анализируемых систем и выделить основные аспекты, которым будет отвечать CMS. При дальнейшем выборе аспектов, мы разделим многообразие CMS на платные и бесплатные. Платные имеют ряд своих плюсов, и один из самых главных — это большая надёжность. Сайты на платных системах управления контентом разрабатываются крупными фирмами, необходимость в надёжности намного выше. Если действительно нужна дополнительная защита от взлома, то платные CMS будут подходить под ваши требования. У нас нет в приоритете большая защищённость, это будет корпоративный сайт и вся информация лежит в открытом доступе. На основе бесплатных CMS можно создать функциональный и удобный по управлению сайт без лишних финансовых расходов.

Определив один из аспектов выбора CMS, отбираем систему по популярности. На сайте <http://www.itrack.ru> в рубрике «Исследования», просмотрев ста-



тистику можно выявить три бесплатных распространенных CMS: WordPress, Joomla, MODx.

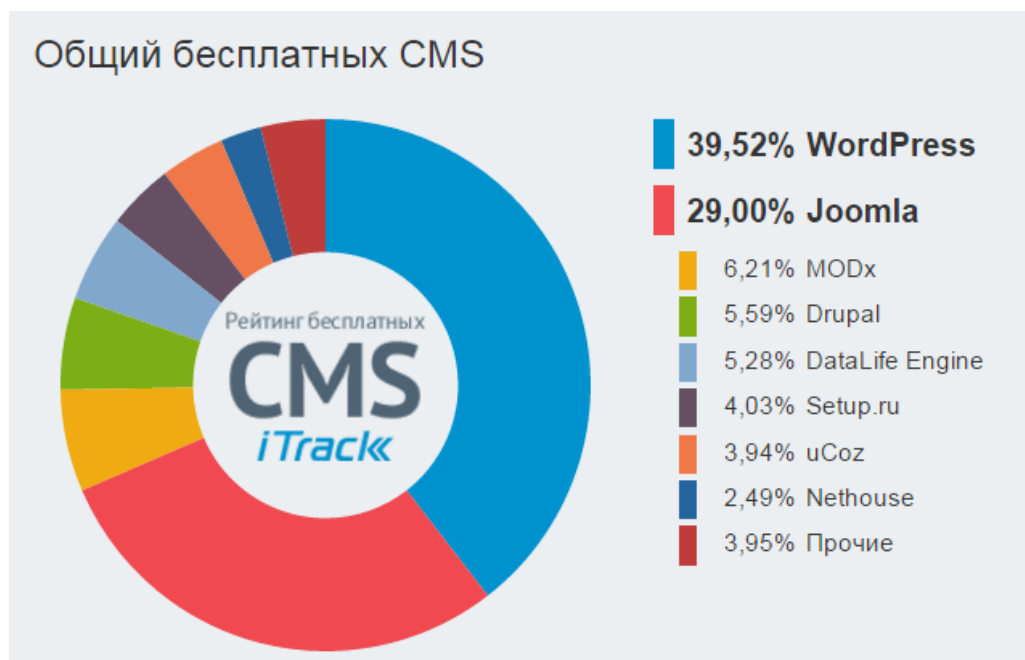


Рисунок 1. Рейтинг бесплатных CMS

Определив список CMS, устанавливаем критерии, по которым можно наиболее объективно оценивать. Выделим важные критерии для разработчика и заказчика для каждой из трех популярных CMS:

- 1) Системные требования к ПК
- 2) Объем дистрибутива
- 3) Системные требование к хостингу
- 4) Простота в обучение – использование CMS на интуитивном уровне
- 5) Набор шаблонов – каталог готовых дизайнов
- 6) Наличие документации
- 7) Поддержка – сообщества, форумы
- 8) Плагины – количество разработок специалистов, расширяющие функциональность CMS
- 9) Виджеты – размещение информации на сайте в отдельные блоги
- 10) Удобство административной части – интуитивно понятная панель администратора

- 11) Обновление
- 12) Безопасность
- 13) Скорость

### 1.2.1 Анализ WordPress

- 1) Для нормализованной работы CMS и плагинов необходимо иметь на компьютере 32-64 Мб оперативной памяти и 50-200 Мб дискового пространства.
- 2) Объем дистрибутива – 4,6 Мб
- 3) Тарифный план – более 100 Мб. Веб-сервер Apache 2.4 и выше, модуль mod\_rewrite, MySQL 5.0 и выше, PHP 5.2.4 и выше.
- 4) WordPress является интуитивно понятной системой управления контентом. Создание страницы, записи и меню происходит в консоли администратора. Выбрав вкладку «Страницы», добавляем новую страницу. Аналогичным способом происходит создание записей.

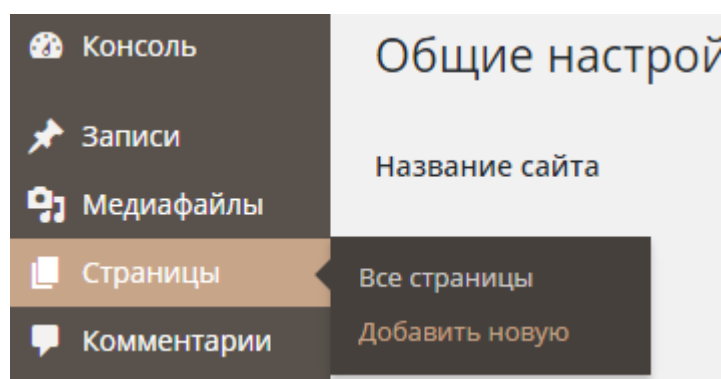


Рисунок 2. Создание новой страницы в WordPress

Выбрав вкладку «Внешний вид», переходим на вкладку «Меню». В настройках меню можно выбрать основные пункты, которые будут отображаться.

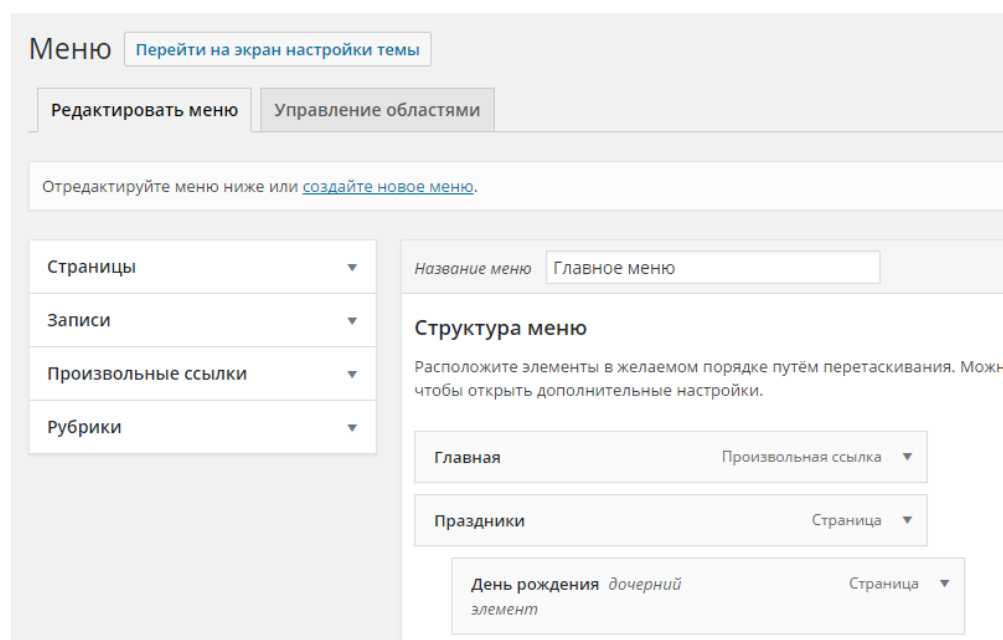


Рисунок 3. Настойка меню в WordPress

Можно сделать вывод, что создание меню, страниц и записей в WordPress не составит большого труда.

- 5) Большое количество шаблонов, сделанных на высоком дизайнерском уровне.
- 6) WordPress имеет большую документацию. Большое количество документации написано на английском языке.
- 7) Есть два русскоязычных сайта. Существует множество статей по описанию различных решений. Проблем в решении вопроса не возникнет. Поддержка пользователей осуществляется через форумы.
- 8) Расширение функциональности WordPress можно осуществлять через плагины. Плагинов для WordPress существует тысячи, таким образом найти нужный плагин под свои цели не составит труда. Есть один недостаток — большинство плагинов написаны на английском языке.
- 9) Виджеты используются для вывода нужного плагина на сайте.

- 10) Административная часть реализована интуитивно понятной, поэтому у пользователей не возникает трудностей при ее освоении. Создать меню или добавить плагин просто и удобно

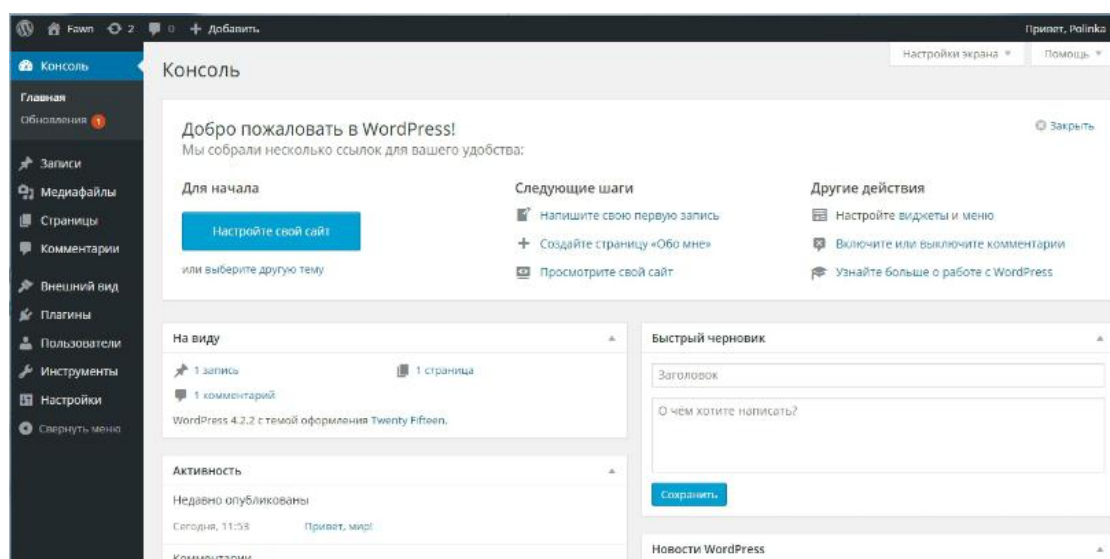


Рисунок 4. Панель администратора WordPress

- 11) Для обновления WordPress необходимо отключить активные плагины. Далее нужно выбрать режим обновления: автоматический или ручной режим. Обновление в автоматическом режиме происходит в вкладке «Консоль» – «Обновления». Рекомендуется использовать автоматический режим обновления версии, так как при таком режиме возникновение ошибки минимальна.

- 12) WordPress является хорошо защищенной платформой. Но существуют уязвимости, которые могут повлиять на безопасность. Необходимо:

- выполнять обновления версии;
- иметь защищенный пароль и логин;
- закрыть обзор директорий;
- сделать редирект с wp-admin на сайт;
- запретить доступ к сайту по ip;

- 13) Каждый раз производится обращение к базе данных. Доступны плагины для кеширования, но настройки ограничены. Скорость при большом количестве плагинов существенно уменьшается.

WordPress идеально подходит для реализации сайта любого проекта. Красочность и многообразие шаблонов, удобство администраторской панели и множество других преимуществ делает WordPress лидером рынка.

### 1.2.2 Анализ MODx

- 1) Для работы необходимо иметь более 64 Мб оперативной памяти и 100-300 Мб дискового пространства.
- 2) Объем дистрибутива – 27,1 Мб
- 3) Тарифный план – более 100 Мб. Веб-сервер Apache 2.2. и выше, IIS 6.0 и выше, Zeus, Cherokee, lighthttpd или nginx; MySQL 5.0 и выше; PHP 5.2.4 и выше. Так же рекомендуется использовать ОС Linux.
- 4) С помощью этой системы можно разработать крупные проекты, порталы и прочее. Для создания страницы нужно нажать на новый документ, после чего заполнить соответствующие поля. Создание страницы в CMS MODx займет больше времени, так как настройка страницы имеет множество пунктов.

Рисунок 5. Создание страницы в MODx

Создание меню необходимо установить расширение – Wayfinder. Расширение Wayfinder служит для организации меню. Для создания меню потребуется большое количество времени, так как расширение Wayfinder имеет множество настроек и нюансов. Проанализировав CMS MODx, можно сделать вывод – система предназначена для опытных пользователей. Чтобы полностью разобраться в функциональности системы, необходимо потратить немало времени.

- 5) Есть готовые шаблоны, но они достаточно однотипны и скучноваты. Дизайнерское решение найти можно, если хорошенько поискать. Если необходим шаблон, отвечающий конкретным, задачам, придется его написать самим.
- 6) Документация MODx есть на русском языке.
- 7) На сайте есть раздел, посвященный ответам на вопросы пользователей. Можно просмотреть самые популярные вопросы, задать свой вопрос, просмотреть теги. Также на сайте есть раздел форума и поддержки. Найти ответ на интересующий вопрос не составит труда.
- 8) Плагинов на MODx не такое большое количество, как на WordPress. Чтобы расширить функциональность CMS, необходимо затратить время на поиски нужного плагина.
- 9) Виджеты отсутствуют. Чтобы реализовать виджеты необходимо создать новый чанк (часть html-кода, используемая в неизменном виде в нескольких местах) и вставить код виджета. Если нужно создать несколько виджетов, следует создать несколько чанков для каждого виджета.
- 10) Достаточно неудобная панель администрирования. Для создания меню потребуются знания в html-программирование. Для реализации меню, страниц и прочего следует прочесть информацию

или уроки. В реализации возникли трудности. Интуитивность отсутствует.

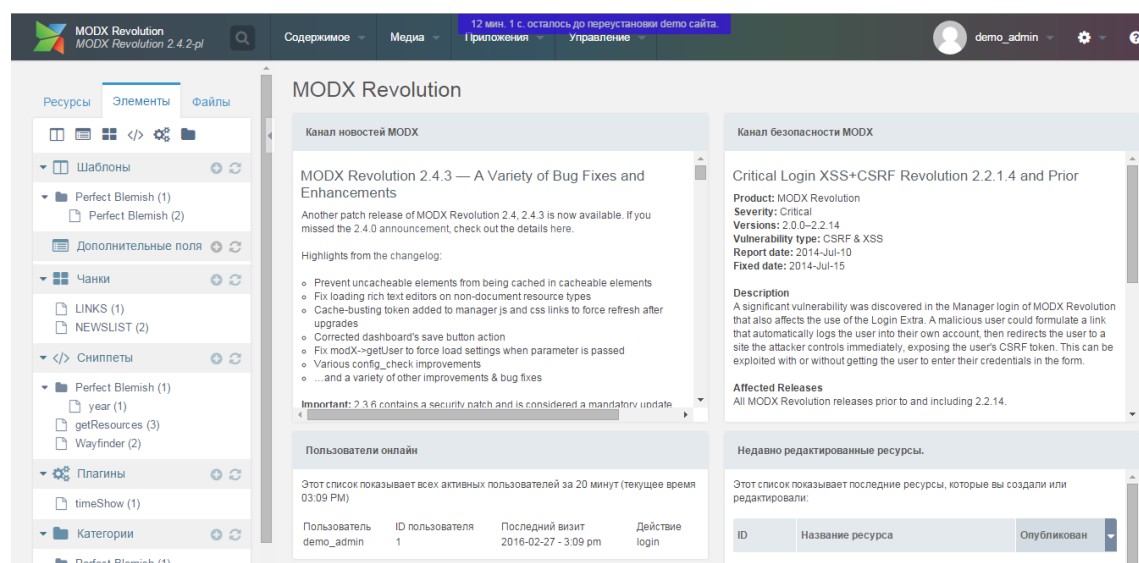


Рисунок 6. Панель администратора MODx

- 11) Нужно быть аккуратней при обновлении версии. Обязательно сделать резервное копирование в случае, если сайт «поломается». При обновлениях могут возникнуть вопросы.

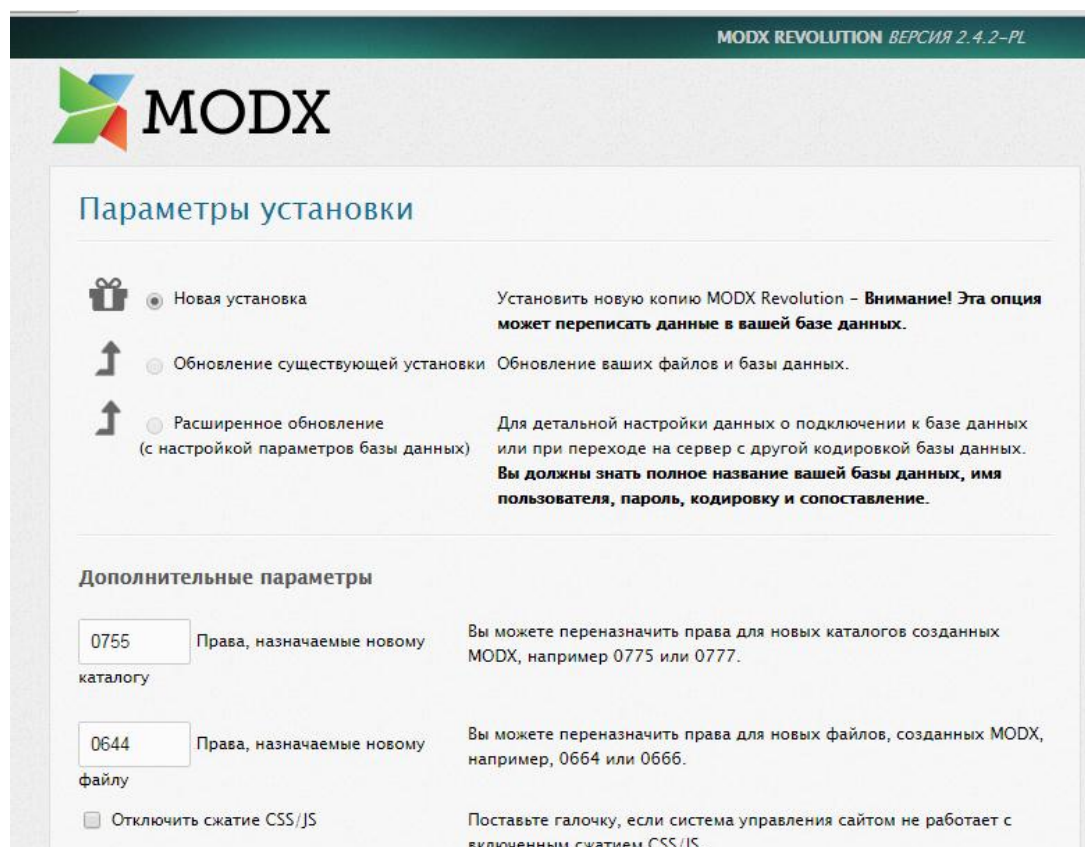


Рисунок 7. Параметры установки обновлений MODx

- 12) MODx является менее распространённой CMS, чем WordPress. В этом есть свой плюс — взломщики меньше уделяют внимания этой системе, ведь большинство сайтов созданы на WordPress или Joomla. Это не особенность программного обеспечения, но является небольшим отличием в безопасности. Так же стоит отметить, что архитектура MODx изначально создавалась с учетом вопросов безопасности.
- 13) Мощный механизм кеширования для снижения нагрузки сервера и поиска базы данных.

MODx подходит для тщательной настройки сайта. Также с помощью данной системы управления контентом можно реализовать сайт любой сложности. MODx предназначен скорее для верстальщиков, чем для неопытных пользователей. При знании программирования на PHP можно без труда реализовывать поставленные задачи.



### 1.2.3 Анализ Joomla

- 1) Для нормализованной работы CMS и плагинов необходимо иметь на компьютере 32-64 Мб оперативной памяти и 50-200 Мб дискового пространства.
- 2) Объем дистрибутива – 26,9
- 3) Тарифный план – более 100 Мб. Веб-сервер Apache 2.2. и выше, PHP 5.3.1.
- 4) На первый взгляд Joomla покажется не удобной. Страница в CMS Joomla называется материал или статья. Для создания статьи необходимо перейти на вкладку «Содержание» – «Статьи» – «Добавить новую». Редактирование статьи имеет множество настроек.

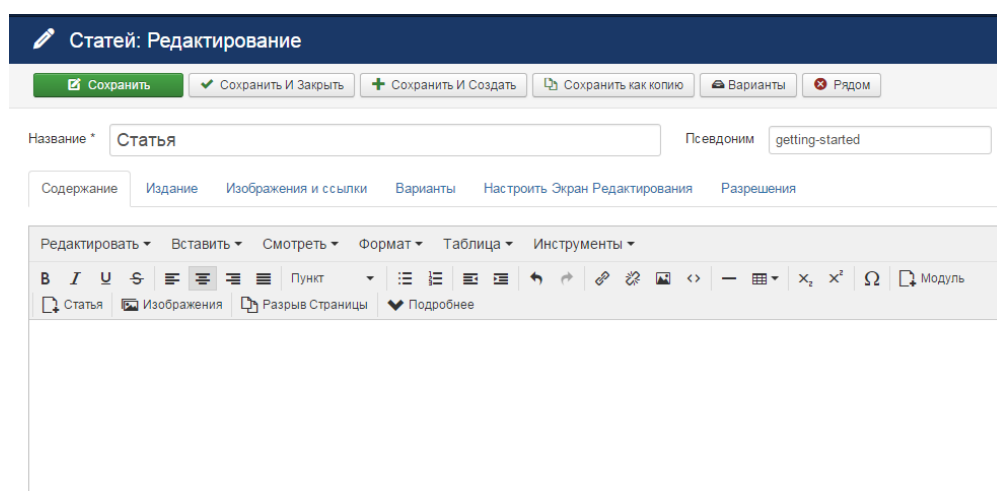


Рисунок 8. Редактирование статьи в CMS Joomla

Для создания меню необходимо перейти во вкладку «Меню» – «Менеджер меню». Меню имеет несколько типов: внутренние ссылки, внешние ссылки, разделитель и псевдоним.

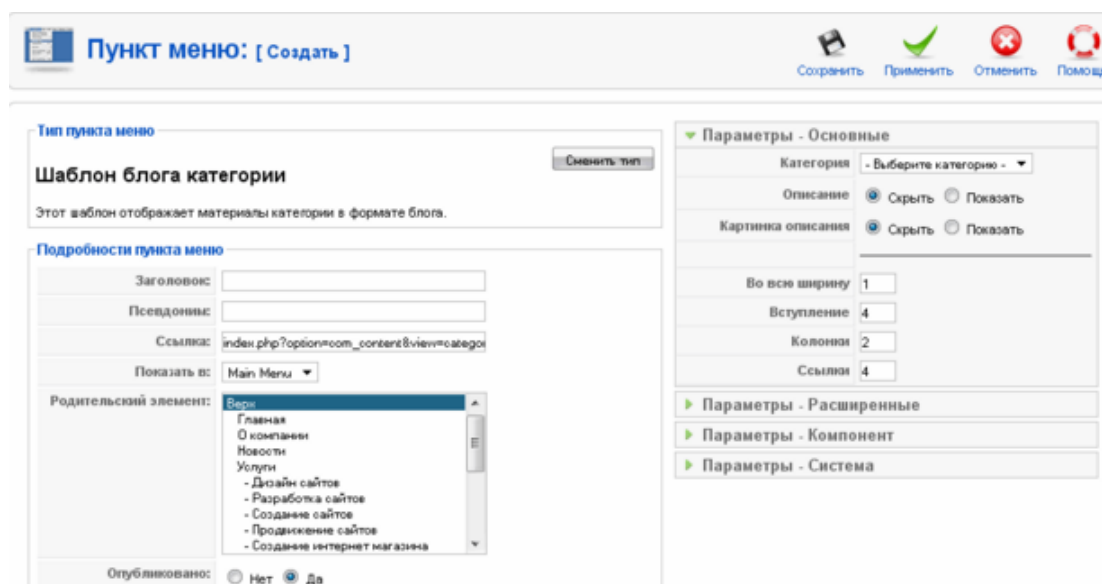


Рисунок 9. Редактирование меню в CMS Joomla

Не интуитивный интерфейс и трудности в использовании могут поставить в ступор неопытных пользователей. Для освоения системы управления контентом Joomla потребуется время.

- 5) Как и WordPress, Joomla обладает огромным количеством шаблонов.
- 6) Документация на русском языке присутствует. Большое количество авторских статей.
- 7) Существует множество сообществ, ведущие свои форумы и сайты поддержки. Трудности с поиском ответа на интересующую проблему не возникнет.
- 8) Плагинов огромное количество, при их активации не возникнет проблем. Присутствует стандартный модуль для авторизации пользователей, самые читаемые статьи и прочее. В некоторых CMS нужно искать специальные плагины, которые могут отвечать за данные функции.
- 9) Виджеты отсутствуют. Необходимо искать специальные модули

- 10) Панель администрирования поначалу может показаться неудобной. Но немного поработав с системой, дальнейшее освоение пойдет проще.

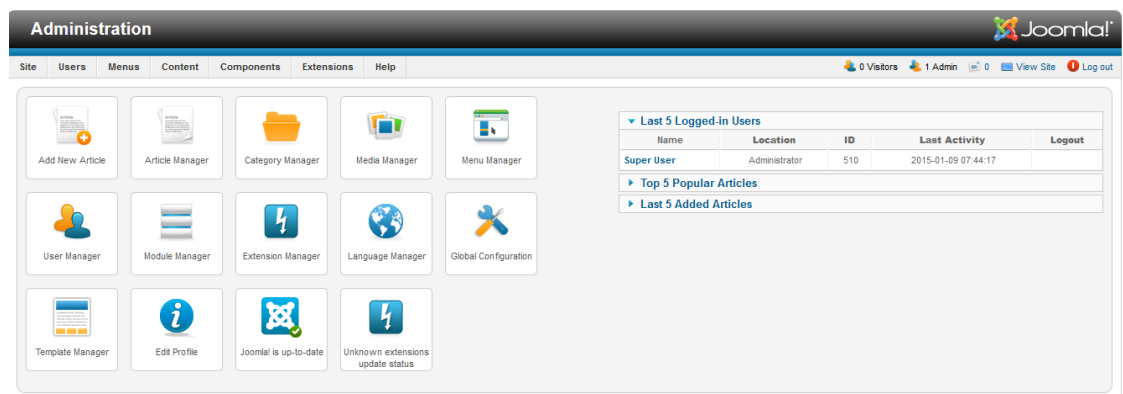


Рисунок 10. Панель администрирования Joomla

- 11) Менеджер расширения позволит обновить устаревшую версию Joomla. Необходимо проверить минимальные требования системы для обновленной версии CMS. Если имеется старая версия Joomla 2.0 до 3 версии, возникнет необходимость чистить кэш сайта, так же необходимо сделать резервную копию.
- 12) Как и любая система, Joomla имеет ряд уязвимостей. При соблюдении стандартных правил по защите CMS, Joomla станет наиболее защищенной.
- 13) Чем больше загружен ваш сайт, тем дольше идет загрузка страницы. На сайтах по Joomla часто можно встретить проблемы касающиеся скорости загрузки страницы или панели администратора. При должной оптимизации можно избежать этих проблем.

Joomla имеет большое количество обучающего материала, отличная система управления контентом для создания корпоративного сайта. Есть несколько недостатков – неудобство работы с шаблонами, запутанная структура, есть необходимость настраивать сторонние компоненты, которые имеют свои особенности, что приводит к сбою работы.

### 1.2.4 Результат и вывод рассматриваемых CMS

Выбор CMS так же основывался на том, что заказчик, получив разработанный сайт, мог с лёгкостью работать в нем, не обладая техническими навыками в предметной области. Это один из немаловажных аспектов при выборе системы управления контентом.

Рассмотрев три популярных CMS, я составила таблицу, на которой представлены категории анализа и оценка данного продукта. Анализ бесплатных CMS производится по 5-ти балльной шкале: 5 – отлично, 4 – хорошо, 3 – удовлетворительно, 2 – плохо.

Таблица 1. Сравнительный анализ CMS

	<b>WordPress</b>	<b>MODx</b>	<b>Joomla</b>
Объем дистрибутива Мб	4,6	27,1	26,9
Простота в обучении	5	2	3
Набор шаблонов	5	3	5
Плагины	5	3	5
Поддержка	5	5	5
Виджеты	5	2	2
Удобство административной панели	5	2	4
Безопасность	4	5	4
Скорость	4	5	3

Исходя из приведенной выше таблицы и сравнительного анализа CMS можно сделать следующие заключение: WordPress подходит для реализации поставленных нами задач. Организации требуется сайт коммерческого характера, имеющий несложную структуру.

## 1.3 Формализованное описание технического задания

### ТЕХНИЧЕСКОЕ ЗАДАНИЕ

Составлен на основе ГОСТ 34.602-89 «Техническое задание на создание автоматизированной системы»

1. Общие сведения.

1.1. Название организации-заказчика.

Организация детских праздников «Таити»

1.2. Название продукта разработки (проектирования).

Веб-сайт

1.3. Назначение продукта.

Веб-сайт разработан для предоставления:

- информации об услугах, новостях и акциях организации
- осуществление обратной связи с клиентами
- демонстрация услуг

1.4. Плановые сроки начала и окончания работ.

Начало разработки – 12 января 2016 г.

Окончание разработки – май-июнь 2016 г.

2. Характеристика области применения продукта.

2.1. Процессы и структуры, в которых предполагается использование продукта разработки.

Разработанный продукт будет применяться заказчиком в коммерческих целях.

2.2. Характеристика персонала (количество, квалификация, степень готовности)

Минимальное количество персонала должно составлять 1 штатная единица – администратор сайта.

Для управления контентом сайта администратор должен обладать общими навыками работы с персональным компьютером и веб-браузером.

3. Требования к продукту разработки.

3.1. Требования к продукту в целом.

- Понятный интерфейс
- Возможность своевременно обновлять информацию на сайте
- Удобное предоставление информации клиентам
- Форма обратной связи
- Интеграция с электронной почтой

3.2. Аппаратные требования.

- Минимальные требования: 1 компьютер с процессором Intel Core i3 и выше, 2x3400 МГц, 2 Гб, 500 Гб, доступ в интернет
- 3.3. Указание системного программного обеспечения (операционные системы, браузеры, программные платформы и т.п.).  
ОС: Windows XP/7/8/10, Linux  
ПО: Любой веб-браузер с включенным интерпретатором JavaScript.  
Поддержка Adobe Flash Player 9 и выше
- 3.4. Указание программного обеспечения, используемого для реализации.  
Для реализации веб-сайта необходимы следующее ПО:  
FTP-клиент FileZilla  
MySQL 5.0 и выше  
PHP 5.2.4 и выше  
CorelDraw Graphics Suite X7  
Photoshop CS6 extended
- 3.5. Для сетевых систем – особенности реализации серверной и клиентской частей.  
ПК администратора должен быть подключен в локальную сеть
- 3.6. Форматы входных и выходных данных  
Входные данные: запрос пользователя по интересующей информации  
Выходные данные: результат запроса пользователя
- 3.7. Меры защиты информации:
- настройка резервного копирования;
  - скрывание версии CMS;
  - защита директорий на сервере от просмотра;
  - надежный пароль для входа в администраторскую часть;
  - обновление версий и плагинов, как только приходит информация об обновлении;
4. Требования к пользовательскому интерфейсу.
- 4.1. Общая характеристика пользовательского интерфейса.
- Удобный и визуально привлекательный сайт;
  - Легкость в ориентировании по сайту;
  - Взаимодействие пользователя с администратором с помощью формы обратной связи;
- 4.2. Размещение информации на экране, дизайн экрана.



Меню содержит кнопки навигации, с помощью которых можно перейти на страницу.

Контент содержит информационное наполнение: текст, графические элементы, медиафайлы.

Подвал содержит информацию об организации, контактные данные.

4.3. Особенности ввода информации пользователем, представление выходных данных.

- При необходимости пользователь может обратиться за консультацией с помощью формы обратной связи;
- При необходимости пользователь может оставить отзыв на веб-сайте организации;

5. Требования к документированию.

Вместе с сайтом должна быть предоставлена документация с инструкцией по работе с администраторским интерфейсом сайта. Документация должна быть представлена заказчику в печатном и электронном варианте.

6. Порядок сдачи-приемки продукта.

Февраль 2016 – разработка структуры сайта, выбор технологии создания

Апрель - Март 2016 – разработка дизайна и функционала сайта

Май 2016 – техническая документация

## Глава 2. Описание процесса разработки веб-сайта

### 2.1 Модельные представления объекта разработки

#### 2.1.1 Этапы проектирование веб-сайта

Перед тем, как разработать веб-сайт, нужно создать понятную и конкретную организацию веб-сайта. Грамотно разработанное проектирование веб-сайта предоставляет возможность логически структурировать работу по реализации и разработке.

На рисунке ниже представлена общая блок-схема по разработке сайта. Она включает в себя 4 ключевых этапа.

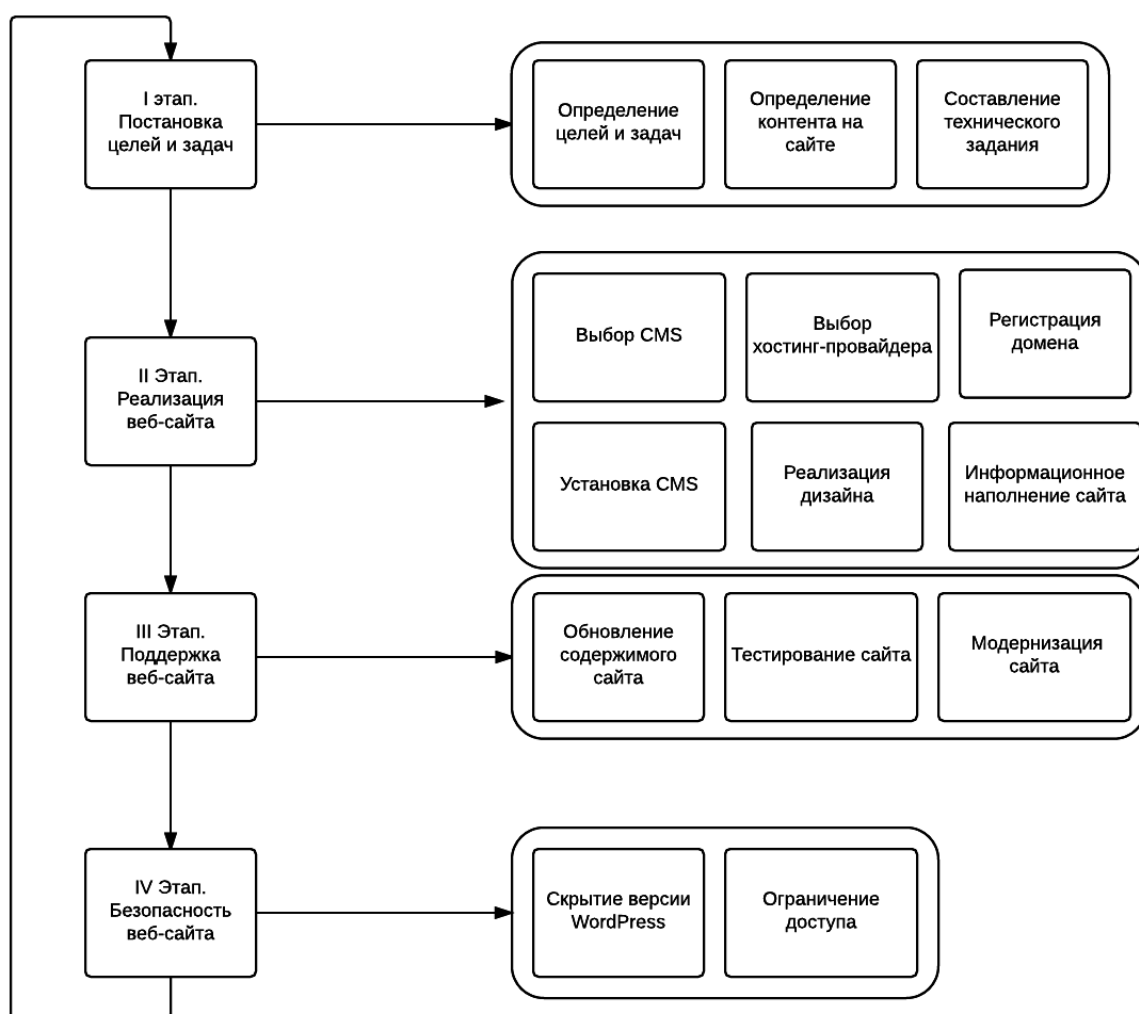


Рисунок 11. Блок-схема этапов проектирования веб-сайта

#### I Этап. Постановка целей и задач веб-сайта



На данном этапе нужно определить, зачем необходим веб-сайт, то есть какие задачи он обязан решать: дать общее представление об организации или проинформировать пользователей, увеличить продажи или организовать веб-торговлю, провести маркетинговую или рекламную кампанию.

Цели и задачи веб-сайта почти всегда должны определяться заказчиком (в нашем случае это будет организация детских праздников «Таити»), а, затем, совместно с исполнителем они устанавливаются и корректируются.

Так же нужно определить контент (содержание) на сайте. Необходимо узнать какую информацию ждут посетители, это может помочь с определением текстовой информацией на веб-сайте.

Еще одним не маловажным пунктом является составление технического задания (ТЗ).

Первый этап – это один из важнейших этапов реализации веб-сайта.

## II Этап. Реализация веб-сайта

Этот этап имеет практический характер. На данном этапе реализуется весь функционал веб-сайта.

После постановки задач и составления технического задания, наступает один из пунктов реализации веб-сайта – это разработка дизайна. Разработка дизайна включает в себя работа над макетом веб-сайта и верстка.

Опираясь на поставленные задачи и техническое задание, разрабатывается макет будущего сайта. На этом этапе определяется, какие графические элементы будут использоваться, как будет выглядеть дизайн в целом, какой цветовой спектр будет использоваться, какая структура будет у страниц. Макет предоставляется в формате PSD для дальнейшей верстки.

Затем происходит «нарезка» готового макета на части и прикрепление этих частей на сайт. Чтобы все элементы на странице отображались в нужном месте, необходимо сверстать веб-страницу. Верстка – это процесс написания HTML и CSS кода. Этот код и отвечает за расположение элементов на странице.

Выбор системы управления содержимым происходит за счет сравнительного анализа популярных CMS.

Затем происходит выбор хостинг-провайдера и регистрация домена. В сети Интернет есть множество хостинг-провайдеров, которые будут предлагать площадки для размещения веб-сайтов. Выбор хостинг-провайдера зависит от множества фактов: от цены и спец предложения до стабильности работы хостинг-провайдера. Регистрация домена может производиться на выбранном хостинге или на любом другом.

Работа сайта и ее эффективность тесно связана с наполнением контента, то есть графическое и текстовое содержимое. Все содержимое сайта должно включать в себя общие концепции веб-сайта и соответствовать целям и задачам. Материалы для наполнения сайта предоставляет заказчик или заказчик может обратиться к копирайтингу.

Дальше происходит интеграция в систему управления контентом. В нашем случае, нет необходимости программировать сайт с «нуля». Интеграция системы управления контентом предоставляет возможность заменить стандартный шаблон, предоставляющийся CMS, на свой собственный разработанный шаблон. Оригинальный шаблон создан на основе реализации своего дизайна.

### III Этап. Поддержка веб-сайта

Дальнейший этап является поддержкой веб-сайта. На этом этапе идет доработка, обновление содержимого сайта, тестирование и корректировка ошибок.

Актуализация информации на сайте (частота обновлений содержимого) является необходимой частью, ведь от этого зависит информированность посетителей веб-сайта.

Тестирование может представлять разнообразные проверки: просмотр веб-сайта в различных браузерах, просмотр сайта при увеличении или уменьшение шрифта, отключение флэш-плеера или JavaScript и многое другое. После

выявление ошибок идет коррекция и модернизация сайта, до тех пор, пока не будут устранены ошибки.

#### IV Этап. Безопасность веб-сайта

Безопасность веб-сайта – довольно важный нюанс в работе системы. CMS WordPress является Open source (открытое программное обеспечение), то злоумышленники, пользуясь открытым исходным кодом, создают уязвимости, троянские вирусы и нередко взламывают веб-сайт. Поэтому для стабильного функционирования веб-сайта необходимо учитывать простые, но эффективные методы защиты веб-сайта.

Каждый этап проектирование связан друг с другом. Для того, чтобы сохранить целостной и структурность веб-сайта, необходимо спроектировать этапы, определить ключевые моменты. Если этого не сделать и «перепрыгивать» с одного этапа на другой, то велика вероятность того, что суть и логика сайта, а также программные и дизайнерские моменты, могут быть утеряны или не соответствовать друг другу. Правильно спроектированный сайт позволяет сохранить целостность веб-сайта, реализовать весь его функционал, а также послужит более эффективной конверсии сайта.

#### **2.1.2 Функциональная модель разработки веб-сайта**

При документировании процесса разработки веб-сайта часто появляется потребность в представлении логических отношений между взаимодействиями функций. Среди большого количества методологий, позволяющие обрисовать бизнес-процессы, IDEF0 (англ. function modeling – методология функционального моделирование) выделяется среди других. IDEF0 модель представляет возможность рассмотреть внутренние функции взаимодействия компонентов.

Рассмотрим подход с использованием IDEF0 на процессе разработки веб-сайта. Согласно с этим подходом, сначала строится контекстная диаграмма, отражающая связь процесса разработки веб-сайта с внешней средой. После того

как модель процесса разработки веб-сайта будет выполнена, необходимо построить декомпозицию функций.

На рисунке представлена функциональная схема процесса разработки веб-сайта.

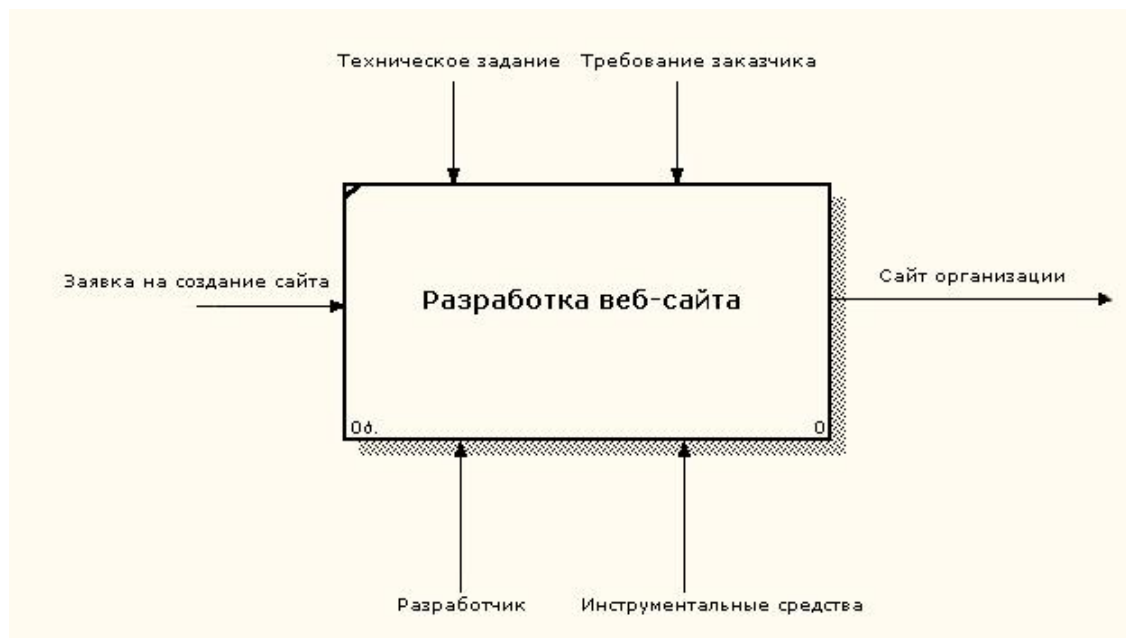


Рисунок 12. Функциональная модель разработки веб-сайта

При построении функциональной модели разработки веб-сайта можно выполнить декомпозицию работ, отраженных на рисунке 8.

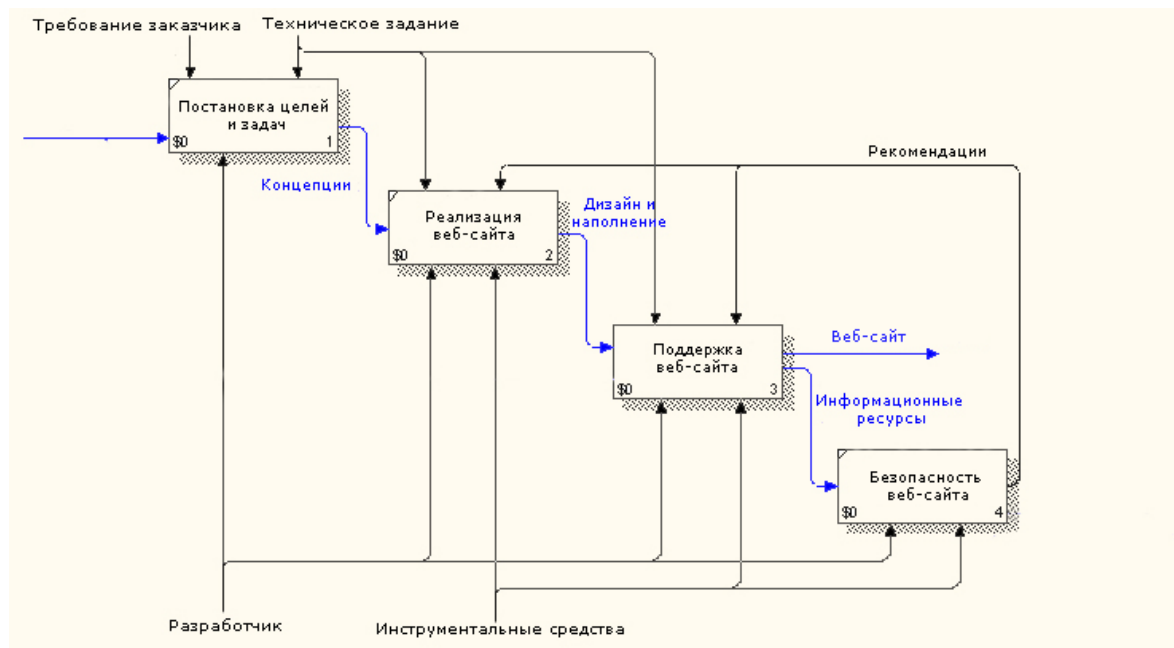


Рисунок 13. Декомпозиция процесса разработки веб-сайта

### 2.1.3 Структура веб-сайта

На рисунке ниже представлена внутренняя структура веб-сайта, которая учитывает всю специфику разработки.

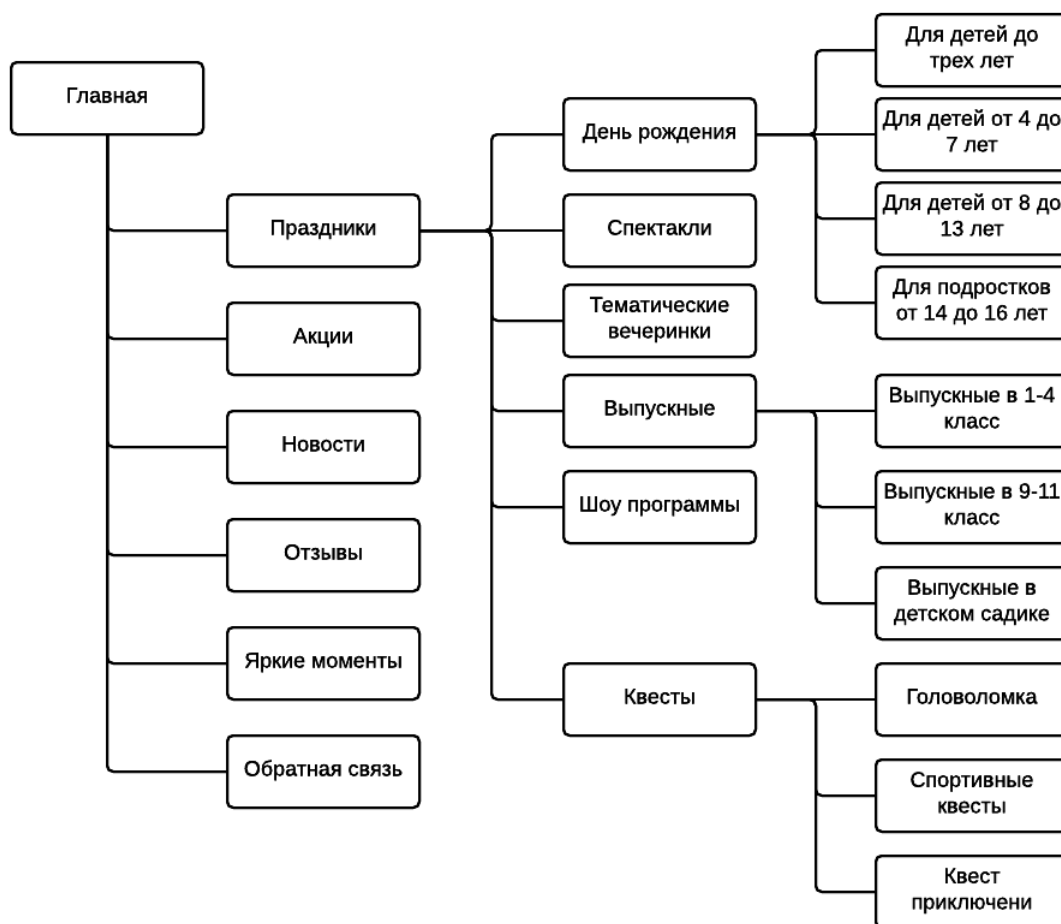


Рисунок 14. Внутренняя структура веб-сайта

На главной странице отображается презентативная информация организации детских праздников. Контент содержит графическую информацию: слайдер и видеоматериал, и текстовую с кратким описанием об организации и описанием ее преимуществ.

На странице «Праздники» представлена информация об услугах организации. Страница «Праздники» содержит следующие разделы: день рождения, спектакль, тематические вечеринки, выпускные, шоу программы, квесты.

Раздел «День рождения» содержит подразделы: для детей до трех лет, для детей от 4 до 7 лет, для детей от 8 до 13 лет, для подростков от 14 до 16 лет. Это позволяет сориентировать посетителя по возрастной категории. После сортировки возрастной категории представляется информация для выбора персонажем на праздник.

Раздел «Спектакли» содержит информацию об услугах проведения спектаклей.

Раздел «Тематическая вечеринка» содержит информацию об организации праздника по определенной тематике.

Раздел «Выпускные» содержит подразделы: выпускные для 1-4 классов, выпускные для 9-11 классов, выпускные для детского сада. Данный раздел включает в себя информацию об услугах на проведения выпускного праздника.

Раздел «Шоу-программы» включает в себя описание праздника в стиле научного шоу.

Раздел «Квесты» содержит в себе подразделы: головоломки, спортивные квесты, квест-приключение. Данный раздел предлагает проведение праздника в стиле квест-игры.

Страница «Акции» содержит информацию о текущих акциях и скидках организации детских праздников.

Страница «Новости» содержит информацию о последних новостях в организации детских праздников «Таити».

Страница «Отзывы» содержит отзывы клиентов.

Страница «Яркие моменты» содержит информацию о проведенных праздниках. Она включает в себя графический материал – фотографии, видео и текстовое описание.

Страница «Обратной связи» содержит форму обратной связи для обращения с вопросом к организации детских праздников.

Отталкиваясь от представления внутренней структуры веб-сайта, можно определить его основные характеристики:

1. Навигация. Важным моментом при разработке веб-сайта является организация интуитивно понятной системой навигации по сайту. Для создания удобной навигации по сайту используются кнопки, гиперссылки и строки меню.
2. Уровни вложенности меню дает представление о необходимом количестве разделов, подразделов и рубрик. Уровень вложенности определяется по минимальному количеству переходов к информации, которая интересует.
3. Контент. Планируется на каждой странице разместить определенные материалы.

После того как определена структура веб-сайта, будет легко представить информационное наполнение каждого раздела. Также опираясь на схему структуры веб-сайта, есть возможность разработать интуитивно-понятную систему навигации.

## **2.2 Описание продукта (результата разработки)**

Разработка веб-сайта предполагает 4 основных этапа. Реализация этапов разработки веб-сайта – это комплекс процессов, которые отвечают за конечный продукт. Краткая характеристика каждого этапа разработки описана ниже.

Первый этап включает в себя:

- постановка целей и задач;
- определение содержание сайта;
- составление технического задания.

Второй этап включает в себя:

- выбор CMS;
- выбор хостинг-провайдера и регистрация доменного имени;
- инсталляция CMS WordPress;
- реализация дизайна;
- информационное напыление веб-сайта.

Третий этап включает в себя:

- обновление содержимого веб-сайта;
- тестирование;
- модернизация веб-сайта;

Четвертый этап включает в себя:

- скрытие версии CMS WordPress;
- ограничение доступа к консоли администратора;
- резервное копирование веб-сайта.

### **2.2.1 Постановка целей и задач**

Для определения целей и задач необходимо ответить на ряд вопросов, которые помогут в дальнейшем с разработкой веб-сайта.

1. Какова аудитория посетителей веб-сайта?
2. Какие функции будет выполнять веб-сайт?
3. Тип веб-сайта.

Так как заказчиком веб-сайта является организатор детских праздников «Таити», то в нашем случае аудиторию посетителей веб-сайта составляют родители или люди, заинтересованные в услугах аниматора. В соответствии с этим к основным функциям веб-сайта относятся:

- Коммуникационная функция (представление информации клиентам);
- Имиджевая функция (позволяет повысить имидж организации);
- Рекламная функция (привлечение клиентов и расширение целевой аудитории);
- Функция обратной связи (представляет форму для консультации клиентов по интересующим вопросам)

Тип веб-сайта позволяет определить, какого рода информация будет преимущественно выделена на веб-сайте. Типом веб-сайта организации детских праздников «Таити» является презентационный веб-сайт. Презентационный



веб-сайт представляет из себя сайт для фирм или организаций малого или среднего бизнеса. На веб-сайтах презентационного плана размещена необходима информация, последние новости, обозначена сфера деятельности и размещается прайс-лист. Основное внимание уделяется уникальному дизайну: стиль оформления, медиа-файлы, фотоматериалы, интерактивность.

Техническое задание описывается в главе 1 пункт 1.3 – Формализованное описание технического задания.

### **2.2.2 Реализация веб-сайта**

После реализации первого этапа, выполняется второй этап, который является основным при разработке.

Проведя в главе 1 1.2 Сравнительный анализ CMS, выбор был остановлен на CMS WordPress. После обсуждения с заказчиком о своем выборе, решение было подтверждено.

Основой стабильной работы веб-сайта является хостинг-провайдер. На сегодняшний день существует множество хостинг-провайдеров и зачастую очень легко запутаться в выборе. Неверный выбор хостинг-провайдера чреват неприятными последствиями, такими как неправильная настройка сервера, что может вызвать проблемы в работе скриптов, а закрытие хостинг-провайдера ведет к потере веб-сайта. О выборе хостинг-провайдера существует большое количество статей. Проанализировав рынок хостинг-провайдеров, я остановила свой выбор на компании timeweb.com. Компания Timeweb отличается стабильной работой сервера, относительно низкими ценами на услуги виртуального хостинга, быстрой и эффективной поддержкой.

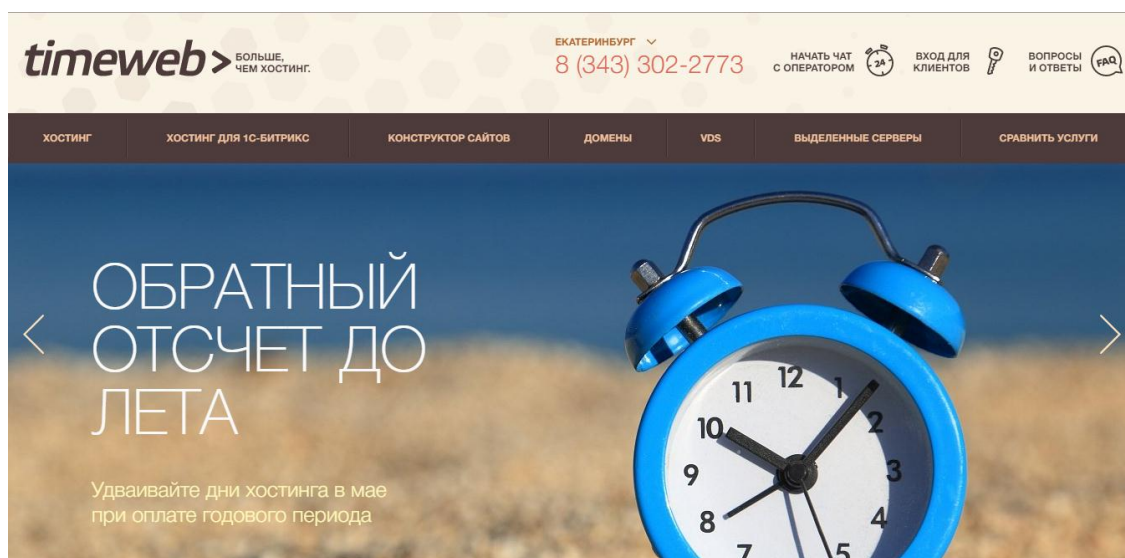


Рисунок 15. Главная страница хостинг-провайдера timeweb.com

Выбор хостинг-провайдера и тарифного плана согласуется и оплачивается с заказчиком. После оплаты тарифного плана высылаются реквизиты на указанную почту. После получения реквизитов доступна cPanel – панель управления веб-хостингом.

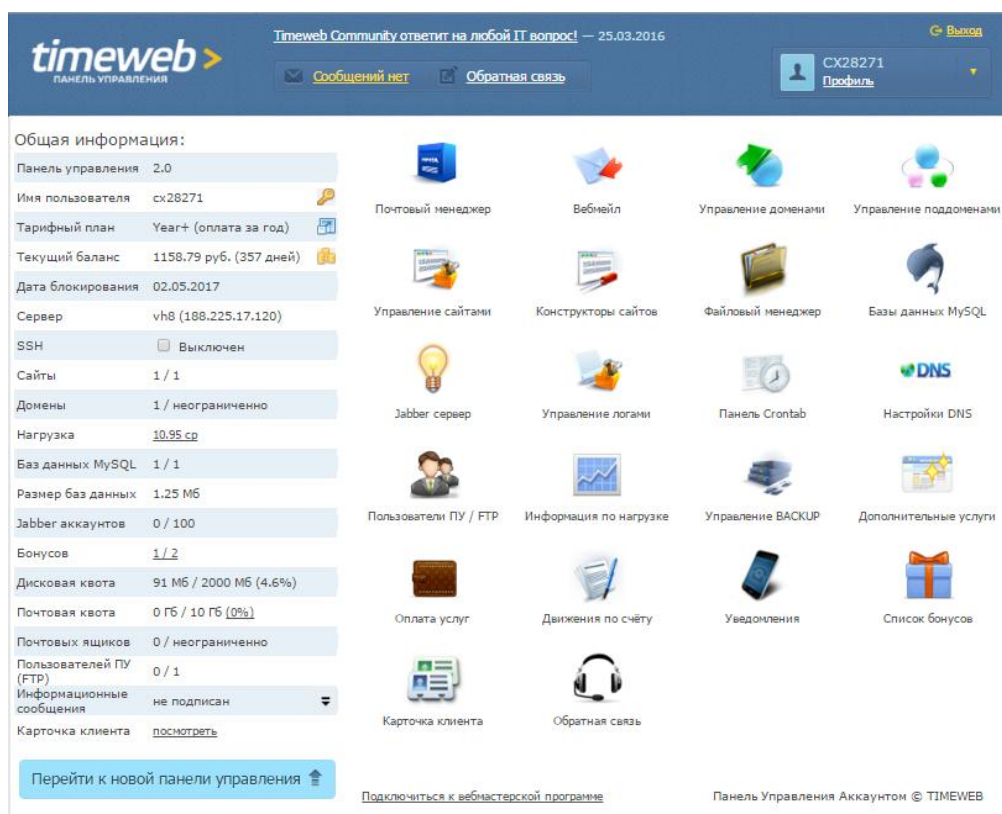



Рисунок 16. cPanel хостинг-провайдера timeweb

В cPanel входит довольно большое количество свободного программного обеспечения, главными из которых являются MySQL, PHP, Apache. Особенно важным программным обеспечением является MySQL и PHP.

После выбора и согласования хостинг-провайдера производится выбор и регистрация доменного имени. Регистрация доменного имени производилась на этом же веб-хостинге. Доменное имя разрабатываемого веб-сайта - <http://taiti-ekb.ru>. Доменное имя имеет два уровня: taiti-ekb – домен второго уровня, .ru – домен первого уровня.

По завершению регистрации доменного имени веб-сайта, наступает реализация следующего пункта – это создание веб-сайта и привязка домена к созданному веб-сайту. В панели управления аккаунтом перейти на вкладку «Управление сайтами» и заполнить необходимые поля.



### Управление сайтами.

В данном разделе Вы можете управлять сайтами на аккаунте: создавать/удалять сайты (мультидомены), привязывать домены к сайту.

Всего сайтов доступно: 1, использовано: 1.

▼ Создание нового сайта (мультидомена)

Укажите директорию, в которой будет расположен новый сайт. Имя директории может состоять из латинских букв, цифр и следующих символов: "-" (тире), "." (точка) и символ "/" (слэш) для указания вложенной директории. Для удобства в качестве директории можете указать имя основного домена, который будет указывать на эту папку.


Директория сайта:

▼ Список существующих сайтов

К одному сайту может быть привязано (прилинковано) несколько доменов. Таким образом, все они будут указывать на одно место на диске. Для привязки домена к сайту, выберите свободный домен, а также необходимую директорию.

Выберите свободный домен для привязки:

Выберите директорию сайта для привязки:

Директория сайта	Домен	Действие
/cx28271.tmweb.ru 		<a href="#">Удалить сайт</a>
<input type="text" value="_____"/>	<a href="#">taiti-ekb.ru</a>	<a href="#">Отлинковать домен</a>

[ [На стартовую страницу](#) ]

Рисунок 17. Создание веб-сайта и привязка доменного имени

По завершению создания веб-сайта, наступает важный момент разработки веб-сайта - инсталляция CMS WordPress. Инсталляция WordPress осуществляется с помощью ftp-клиента. Так же можно воспользоваться файловым менеджером (WinSCP, Total Commander, FAR и др.) или специализированной программой FileZilla. Я выбрала ftp-клиент FileZilla и обосновываю свой выбор тем, что он удобен в использовании и это открытое программное обеспечение. Для начала работы по инсталляции системы управления контентом необходимо загрузить программу FileZilla с официального сайта <http://filezilla.ru/> и установить на ПК, где будет размещаться ftp-сервер. После установки FileZilla запускаем программу и вводим данные веб-сайта. Данные для ftp-подключения высылаются на почту хостинг-провайдером. Нужно ввести данные хоста, имя пользователя и пароль (порт указывать необязательно).

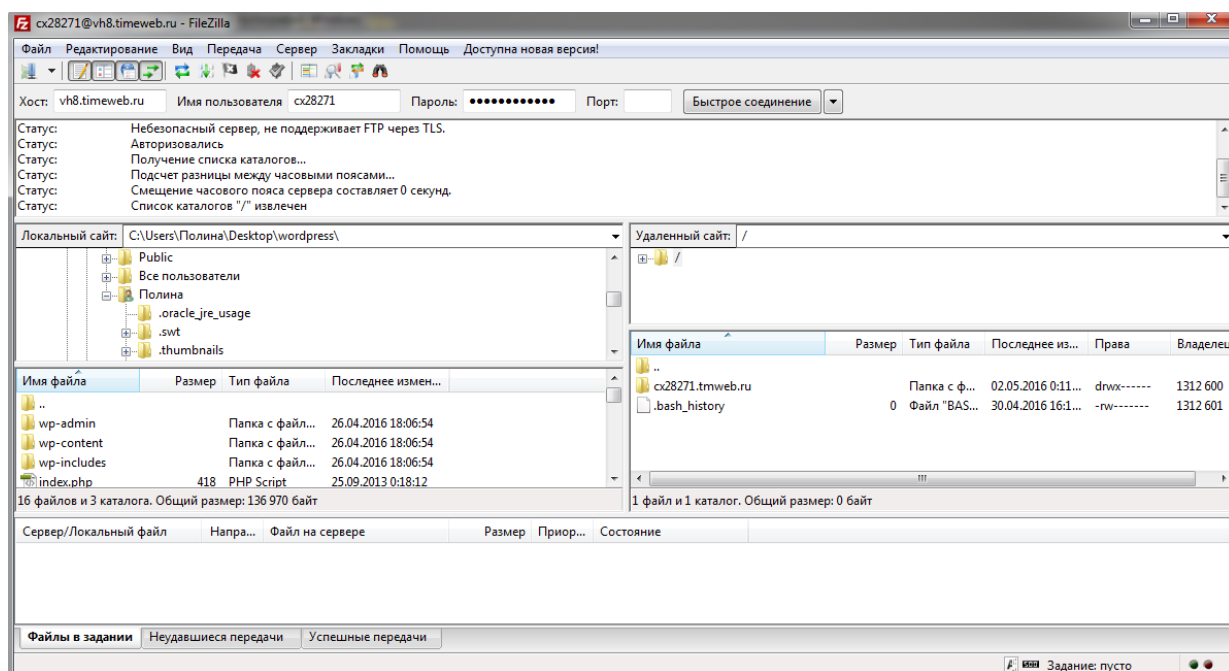



Рисунок 18. Окно программы FileZilla


Для загрузки системы управления контентом WordPress, необходимо скачать дистрибутив с официального сайта <http://ru.wordpress.org/>. При помощи ftp-клиента необходимо скопировать полученные файлы CMS в корневую директорию сервера public\_html.

После того как файлы WordPress установлены на сервер, к нему необходимо подключить базу данных, но сначала ее надо создать. Следующим шагом установки WordPress является создание базы данных MySQL и изменение некоторых параметров, необходимых для доступа к этой базе данных. MySQL – одна из самых популярных и распространенных баз данных с открытым исходным кодом. Она используется многими хостинг-провайдером, timeweb также ее использует. Для создания базы данных MySQL необходимо перейти в панель управления веб-хостингом, выбрать «Базы данных MySQL» и заполнить соответствующие поля.



## Управление базами данных MySQL.

В данном разделе Вы можете создавать/удалять базы MySQL, редактировать доступ к базам с внешнего IP, менять пароли на базы.



Полный доступ включен.

Включение полного доступа добавляет ряд удобств:

- в phpMyAdmin вы увидите список всех баз данных на вашем аккаунте и список этих баз будет увеличиваться при создании дополнительных баз данных;
- вы сможете входить в phpMyAdmin по ссылке ниже, не вводя при этом пароль.

Войти в phpMyAdmin



Выключить полный доступ

Всего баз данных доступно: 1, использовано: 1.

Укажите желаемое имя базы данных и пароль. База будет создана автоматически.

Имя базы: cx28271\_  Пароль: 

Создать

База данных/Пользователь	Доступ из		
cx28271_polina (phpMyAdmin)	<input type="text"/>	<a href="#">Добавить доступ</a>	
	localhost	<a href="#">Изменить пароль</a>	

**Полезная информация о MySQL:**

1. <http://mysql.com> - Основной сайт разработчиков (eng)
2. <http://dev.mysql.com/doc/refman/5.0/en/index.html> - Оригинальная документация для версии 5.0.x (eng)

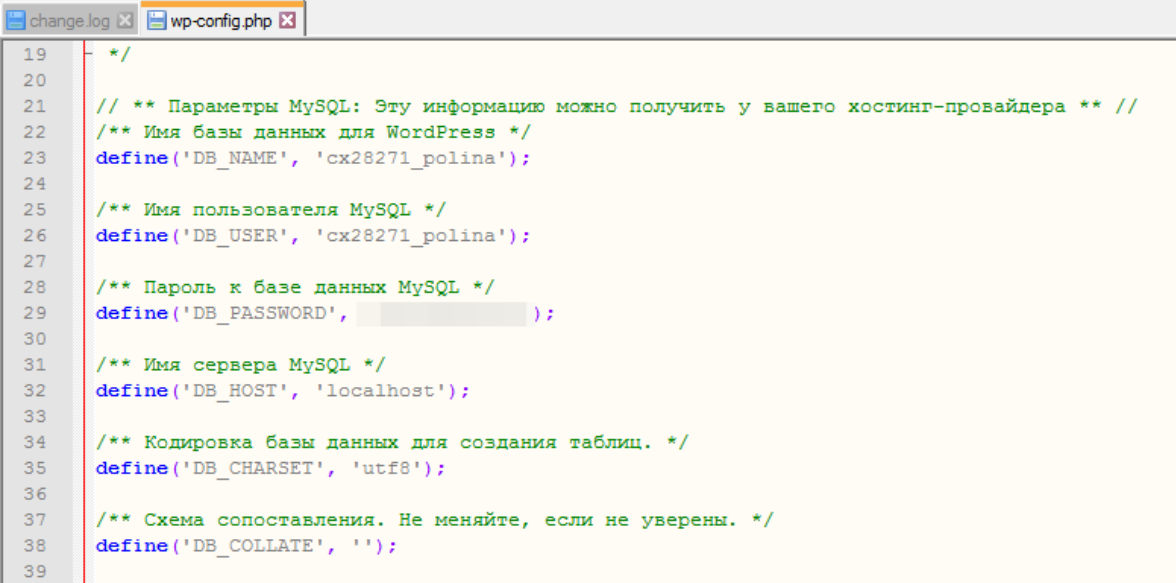
(Все ссылки открываются в новых окнах, данная информация представлена для ознакомления)

[ [На стартовую страницу](#) ]

Рисунок 19. Создание базы данных MySQL

В файлах, загруженных на ftp-клиенте, находим файл `wp-config-sample.php` и переименовываем `wp-config.php`. Открываем этот файл в любом текстовом редакторе и редактируем следующие поля:

- `DB_NAME` – название базы данных;
- `DB_USER` – пользователь базы данных;
- `DB_PASSWORD` – пароль для пользователя;



The screenshot shows a text editor window with two tabs: 'change.log' and 'wp-config.php'. The 'wp-config.php' tab is active, displaying the following PHP code:

```
19  */
20
21  // ** Параметры MySQL: Эту информацию можно получить у вашего хостинг-провайдера ** //
22  /** Имя базы данных для WordPress */
23  define('DB_NAME', 'cx28271_polina');
24
25  /** Имя пользователя MySQL */
26  define('DB_USER', 'cx28271_polina');
27
28  /** Пароль к базе данных MySQL */
29  define('DB_PASSWORD', ' ');
30
31  /** Имя сервера MySQL */
32  define('DB_HOST', 'localhost');
33
34  /** Кодировка базы данных для создания таблиц. */
35  define('DB_CHARSET', 'utf8');
36
37  /** Схема сопоставления. Не меняйте, если не уверены. */
38  define('DB_COLLATE', '');
39
```

Рисунок 20. Подключение базы данных MySQL

Для завершения установки WordPress необходимо запустить скрипт установки WordPress. В панели управления аккаунтом переходим на вкладку «Конструктор сайтов», внизу в выпадающем списке «Список установленных сайтов» и выбираем «реквизиты». В окне «параметры установленной системы» находим ссылку «точку входа» и переходим на скрипт установки WordPress.

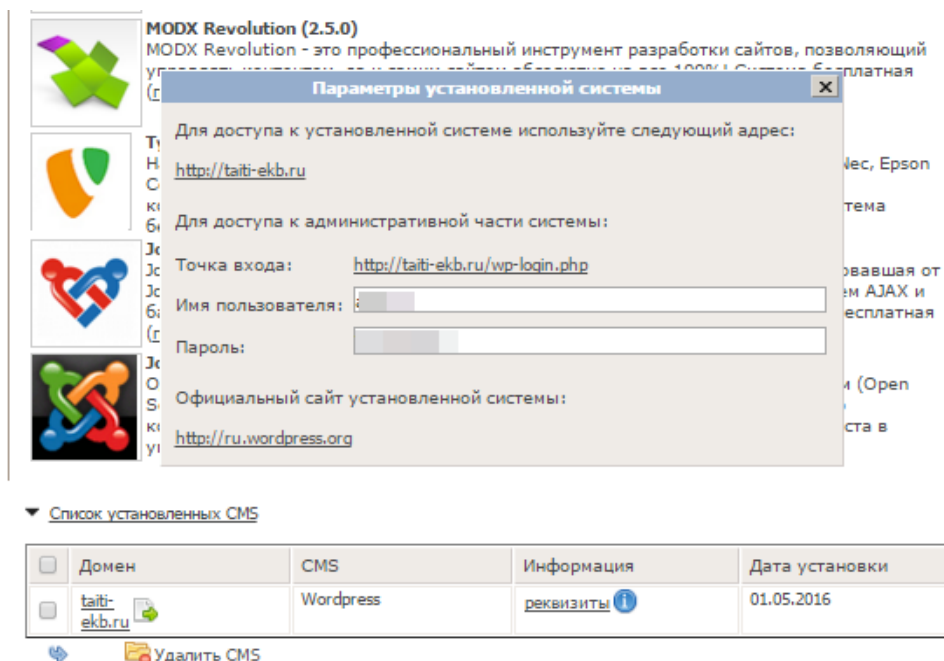


Рисунок 21. Окно «Параметры установленной системы»

Если инсталляция прошла успешно, то автоматически переходим на форму входа в консоль администратора.

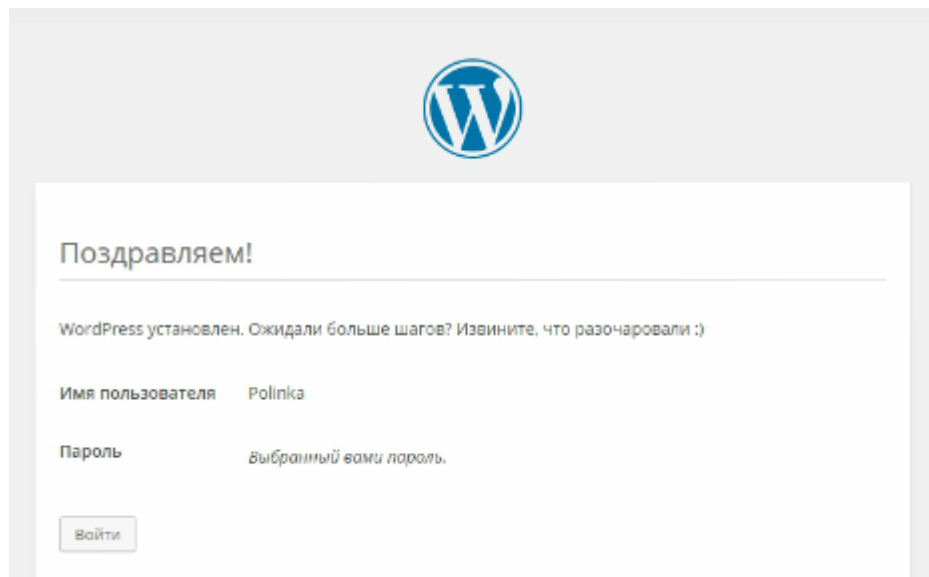


Рисунок 22. Форма, информирующая об успешные установки WordPress

Инсталляции системы управления контентом завершена, переходим к следующему пункту разработки – это реализация дизайна. Реализация дизайна включает в себя макет веб-сайта и верстку.



WordPress предлагает множество шаблонов с красочным дизайном для веб-сайта, но нам требуется свой собственный дизайн для организации детских праздников «Таити» с учетом стилевых характеристик.

Для начала необходимо определиться с программным обеспечением для реализации дизайна. Я выбрала следующие программное обеспечение: CorelDraw Graphics Suite X7 и Photoshop CS6 extended. С помощью CorelDraw Graphics Suite X7 была реализована графическая часть: логотип организации, фоновое изображение, а также подобран цветовой спектр.

В Photoshop CS6 extended был реализован макет. Разметка макета позволяет сориентироваться с размерами веб-сайта. Это поможет «сверстать» страницу качественно.

В результате с помощью программного обеспечения был продуман и реализован макет веб-сайта организации.

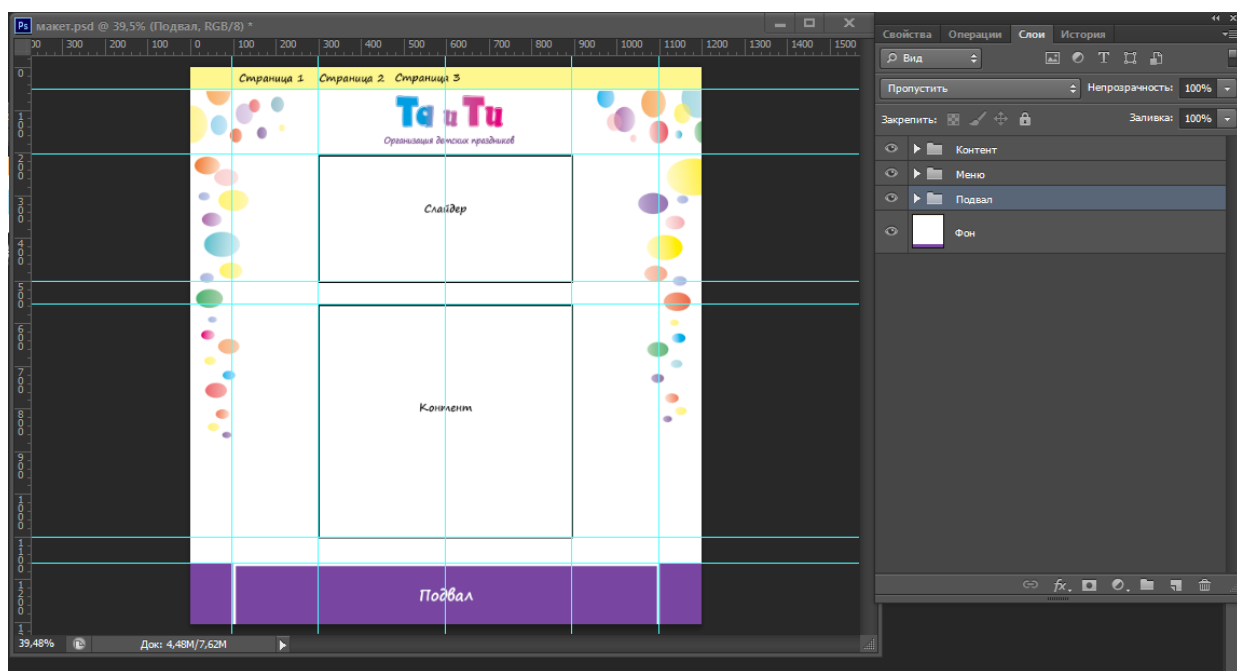


Рисунок 23. Макет веб-сайта организации

Затем происходит верстка готового макета и прикрепление этих частей на веб-сайт. Верстка – это формирование веб-сайта по готовому макету. Этот процесс включает в себя создание кода страницы при помощи языка разметки гипертекста HTML и оформление ее с помощью каскадных таблиц стилей CSS.



Чтобы «сверстать» веб-сайт, нужно знать «анатомию» системы управления контентом.

Имя файла	Размер	Тип файла	Последнее из...	Права	Владелец/...
wp-admin		Папка с ф...	10.05.2016 13:3...	drwx-----	1312 600
wp-content		Папка с ф...	10.05.2016 13:4...	drwx-----	1312 600
wp-includes		Папка с ф...	02.05.2016 0:08...	drwx-----	1312 600
.htaccess	235	Файл "НТ...	02.05.2016 0:16...	-rw-----	1312 600
index.php	418	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
wp-activate.php	5 032	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
wp-blog-header.php	364	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
wp-comments-post.php	1 476	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
wp-config-sample.php	4 179	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
wp-config.php	4 260	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
wp-cron.php	3 286	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
wp-links-opml.php	2 380	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
wp-load.php	3 316	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
wp-login.php	33 837	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
wp-mail.php	7 887	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
wp-settings.php	13 106	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
wp-signup.php	28 624	PHP Script	02.05.2016 0:16...	-rw-----	1312 600
16 файлов и 6 каталогов. Общий размер: 115 496 байт					

Рисунок 24. Корневой каталог WordPress

Корневой каталог WordPress содержит три основных папки: wp-admin (папка содержит файлы, которые обеспечивают работу консоли и административной части), wp-content (папка содержит все загруженные пользовательские данные), wp-includes (папка содержит все необходимые данные для запуска WordPress через пользовательский интерфейс). Папки wp-admin и wp-includes являются основными, в них желательно ничего менять. Для верстки сайта понадобится папка wp-content, которая в свою очередь содержит такие папки как languages, plugins, themes, uploads.

Папка themes отвечает за загруженные шаблоны тем. В папке themes имеется множество файлов с расширением php. Основные файлы шаблона, с которыми предстоит работать:

- style.css – таблица стилей шаблона, реализует оформление;
- index.php – файл, отвечающий за вывод главной страницы;

- search.php – этот файл отвечает за отображение результатов поиска на веб-сайте;
- 404.php – страница ошибки 404;
- comments.php – файл для отображения комментариев на страницах;
- footer.php – файл отвечает за отображение подвала веб-сайта;
- header.php – файл отвечает за вывод шапки сайта;
- sidebar.php – файл выводит боковые колонки;
- page.php – файл, который отображает структуру страницы;
- functions.php – файл, где прописаны функции шаблона.

После ознакомления с файлами шаблона, нужно сверстать шаблон веб-сайта из PSD-формата в HTML и CSS. Файлы шаблона можно взять из стандартной темы WordPress, она устанавливается автоматически сразу после того как произведена инсталляция CMS. Каждый шаблон имеет ту же архитектуру, которая была рассмотрена, только местами меняется HTML и CSS код, так как у каждой темы разное расположение элементов веб-сайта и оформление.

С помощью программного обеспечения Photoshop CS6 была произведена «нарезка» частей макета. Будущий шаблон веб-сайта делиться на три части: логотип, фоновое изображение и подвал. Полученные части макета с помощью ftp-клиента загружаем в папку «images».

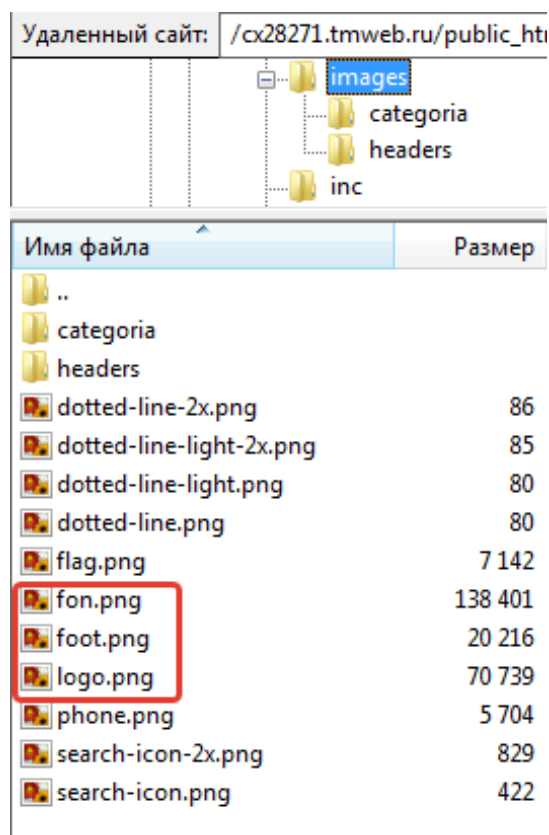


Рисунок 25. Загрузка частей макета в папку шаблона

После того, как все файлы макета были загружены, необходимо определить место расположения каждого элемента. Для этого нужно написать HTML-код и CSS-код. Сначала напишем код, чтобы отобразился логотип организации. Открываем файл header.php с помощью любого текстового редактора и пишем следующий HTML-код.

```
<div id="page" class="hfeed site">

<div id="header">

<?php if(function_exists('wp_content_slider')) { wp_content_slider(); } ?>
<header id="masthead" class="site-header" role="banner">

</header><!-- #masthead -->
```

Рисунок 26. HTML-код для вставки изображения логотипа организации

Этот код отвечает за загрузку изображения логотипа в шапке веб-сайта. Также в нашем макете указано, что меню находится на верху страницы. Для того, чтобы меню на странице отображалась верхней части страницы, находим тег с прописанными внутри атрибутами `<div id="navbar" class="navbar">` и пе-

ремешаем его до тега <body>. На скриншоте ниже представлен результат действия.

```
<body <?php body_class(); ?>>

    <div id="navbar" class="navbar">
        <nav id="site-navigation" class="navigation main-navigation" role="navigation">
            <button class="menu-toggle"><?php _e( 'Menu', 'twentythirteen' ); ?></button>
            <a class="screen-reader-text skip-link" href="#content" title="<?php esc_attr_e(
'Skip to content', 'twentythirteen' ); ?>"><?php _e( 'Skip to content', 'twentythirteen' ); ?></a>
            <?php wp_nav_menu( array( 'theme_location' => 'primary', 'menu_class' => 'nav-menu',
'menu_id' => 'primary-menu' ) ); ?>
        </nav><!-- #site-navigation -->
    </div><!-- #navbar -->
```

Рисунок 27. HTML-код переноса меню в верхнюю часть страницы

Дальше нужно загрузить фоновое изображение страницы веб-сайта. Надо отметить, что размер изображения должен совпадать с размерами веб-сайта, иначе есть вероятность того, что изображение будет выглядеть не так как планировалось.

За реализацию фонового изображение отвечает файл style.css. Открываем файл и находим нужный класс, который отвечает за вывод фонового изображения. Для определения какой класс отвечает за расположения элементов на странице, можно воспользоваться функцией «просмотр кода». Данная функция действует на многих популярных браузерах и воспользоваться ей можно с помощью «клика» правой кнопкой мыши. Класс, отвечающий за загрузку фонового изображения, является класс hentry. На скриншоте ниже представлен код для вставки фонового изображения.

```
.hentry {
    background: url(images/fon.png);
}
```

Рисунок 28. CSS-код вставки фонового изображения

Таким же принципом происходит поиск класса и написания CSS-кода для загрузки изображение подвала.

```
.site-footer .widget-area,
.sidebar .site-footer {
    text-align: left;
    background: url(images/foot.png)
}
```

Рисунок 29. CSS-код вставки изображения подвала веб-сайта

После загрузки частей макета, необходимо настроить навигацию веб-сайта. В структуре веб-сайта были описаны разделы, которые будут реализованы. Чтобы создать меню сначала создадим страницы веб-сайта. В административной консоли WordPress находим раздел «Страницы» - «Добавить новую».

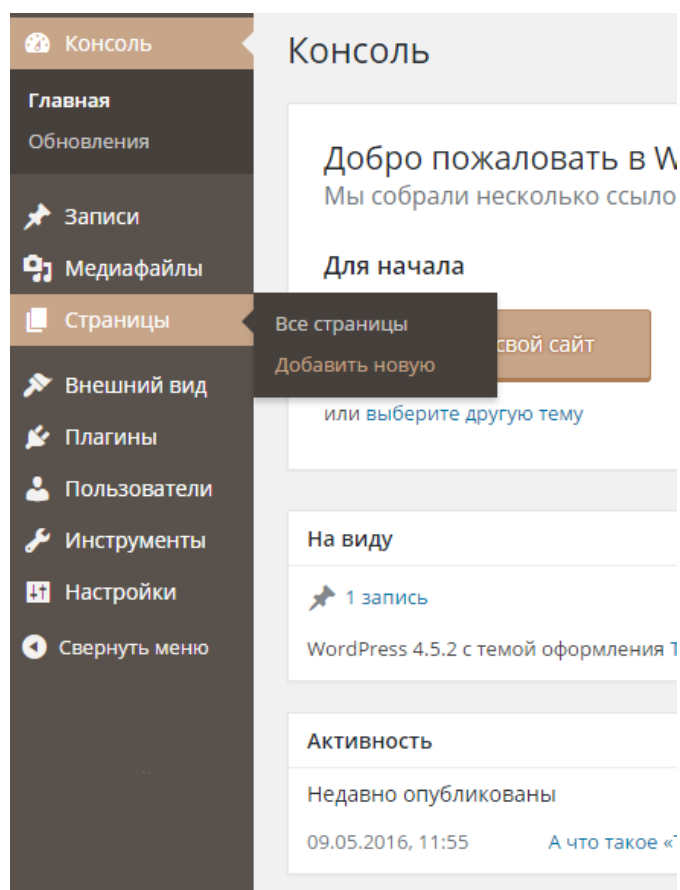


Рисунок 30. Создание новой страницы

В открывшемся редакторе страниц для настройки меню в «атрибуты страницы» указываем родительскую страницу. Это создаст вложенность меню. Когда все страницы созданы, должна получиться иерархия страниц.

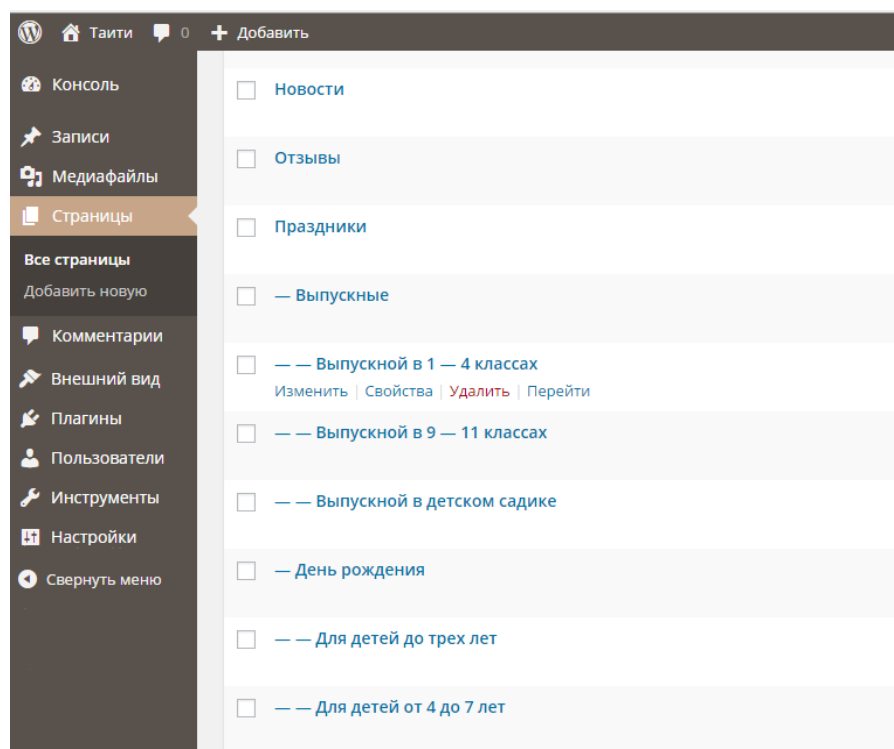


Рисунок 31. Структура страниц веб-сайта

Создание меню в CMS WordPress происходит следующим образом: в разделе «Внешний вид» выбираем «Меню» и перетаскиваем поля, которые нужно отобразить.

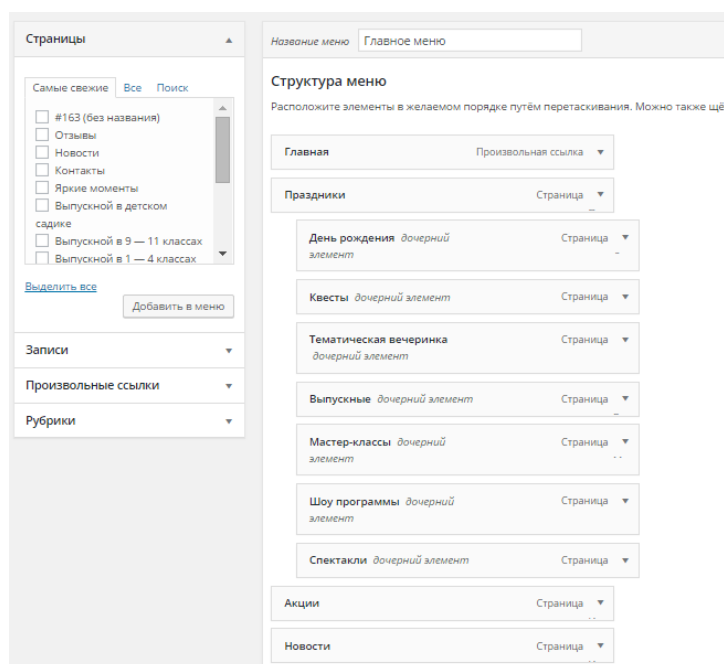


Рисунок 32. Настройка навигации веб-сайта

В макете веб-сайта был отмечен важный момент – это слайдер. На сегодняшний день практически каждый веб-сайт имеет слайдер. Слайдер представляет галерею изображений, которая автоматически переключается или переключение происходит с помощью кнопок. Слайдер делает веб-сайт ярким и интересным, также делает акцент на графической информации. Для организации детских праздников это является эффективным способом демонстрации своих услуг. Существует большое количество плагинов для реализации слайдера веб-сайта. Плагин – это программный модуль, который подключается к основной программе и расширяет функциональность. Установка плагина в WordPress не составляет большого труда. Для установки плагина нужно перейти в раздел «Плагин» и добавить новый. Дальше вводим название нужного плагина в поисковой строке. В нашем случае, выбор плагина для слайдера стал плагин «Easing Slider».

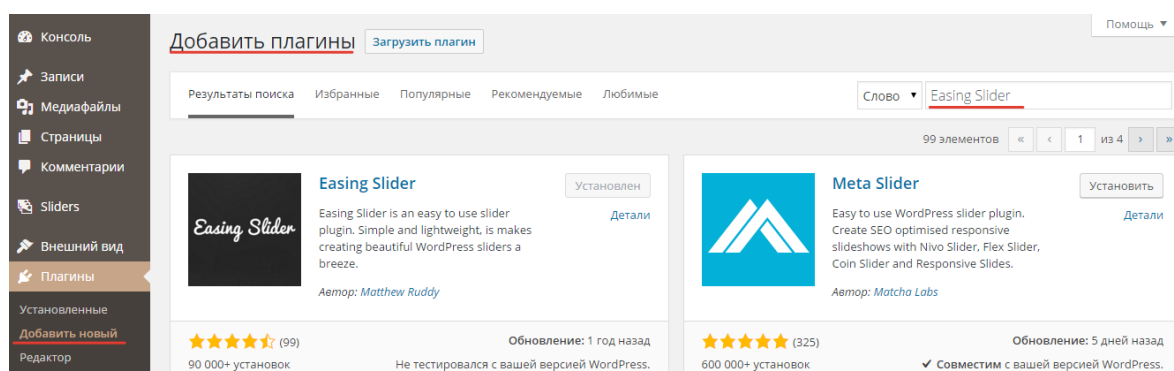


Рисунок 33. Вкладка «Добавить плагин»

После активизации плагина «Easing Slider» в консоли появиться новая вкладка «Slider». Добавить новый слайдер можно с помощью вкладки «Add Sliders». Фотоматериал, который будет отображен на главной странице веб-сайта организации, предоставляется самой организацией. После добавления изображений плагин автоматически генерирует php код и shortcode.

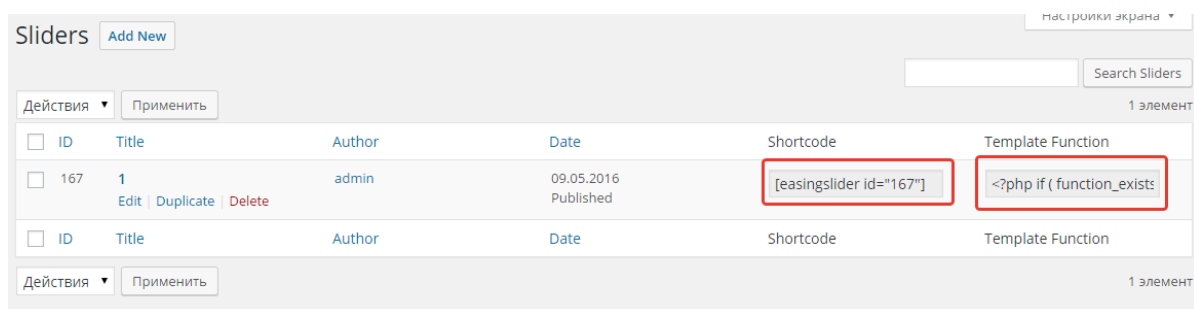


Рисунок 34. Готовый слайдер

Shortcode – это набор функций, создающие небольшой код, который можно использовать в редакторе страниц, записей, виджитах и в файлах шаблона. В редактора страниц вставляем shortcode на главную страницу веб-сайта.

В результате проделанной работы по реализации дизайна веб-сайта организации получаем готовый шаблон к наполнению информационным содержанием.



Рисунок 35. Главная страница веб-сайта организации

Также необходимо реализовать форму обратной связи посетителя с организацией. Большое число плагинов ведет к повышенной нагрузки веб-сайта. Чтобы не перегружать веб-сайт было предложено написать код для реализации формы обратной связи. Создаем страницу «Обратная связь» и вставляем HTML-код.



```

<form method="POST" id="feedback-form">
  Как к Вам обращаться:
  <input type="text" name="nameFF" required placeholder="Ваше имя" x-
  Email для связи:
  <input type="email" name="contactFF" required placeholder="Адрес эл
  Ваш номер телефона:
  <input type="text" name="phoneFF" required placeholder="Например: +
  Ваше сообщение:
  <textarea name="messageFF" required rows="5"></textarea>
  <input type="submit" value="отправить">
</form>

```

Рисунок 36. HTML-код формы обратной связи

Так же необходимо создать php-файл и вставить код представленный на рисунке 36. Этот код реализует доставку сообщения на почту администратора.

```

<?php
$post = (!empty($_POST)) ? true : false;
if($post) {
    $email = $_POST['email'];
    $name = $_POST['name'];
    $email = $_POST['email'];
    $sub = $_POST['sub'];
    $message = $_POST['message'];
    $error = '';
    if(!$name) {$error .= 'Укажите свое имя. '};
    if(!$email) {$error .= 'Укажите электронную почту. '};
    if(!$sub) {$error .= 'Укажите тему обращения. '};
    if(!$message || strlen($message) < 1) {$error .= 'Введите сообщение. '};
    if(!$error) {
        $address = "polinko_oo@mail.ru";
        $mes = "Имя: ".$name."\n\nТема: " . $sub."\n\nСообщение: ".$message."\n\n";
        $send = mail ($address,$sub,$mes,"Content-type:text/plain; charset = UTF-8\r\nFrom:$email");
        if($send) {echo 'OK!';}
    }
    else {echo '<div class="err">'.$error.'</div>';}
}
}
?>

```

Рисунок 37. Код для реализации отправки сообщения на почту

Для реализации самого алгоритма заполнения формы, необходимо создать js-файл или подключить скрипт в самом HTML-документе.

```

<script>
document.getElementById('feedback-form').addEventListener('submit', function(evt){
    var http = new XMLHttpRequest(), f = this;
    evt.preventDefault();
    http.open("POST", "/wp-content/themes/twentythirteen/mail.php", true);
    http.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
    http.send("nameFF=" + f.nameFF.value + "&contactFF=" + f.contactFF.value + "&phoneFF=" + f.phoneFF.value + "&messageFF=" + f.messageFF.value);
    http.onreadystatechange = function() {
        if (http.readyState == 4 && http.status == 200) {
            alert(http.responseText + ', Ваше сообщение получено.\nМы ответим Вам в течении 2-х дней.\n');
            f.messageFF.removeAttribute('value'); // очистить поле сообщения (две строки)
            f.messageFF.value='';
            location.href='http://taiti-ekb.ru/';
        }
    }
    http.onerror = function() {
        alert('Извините, данные не были переданы');
    }
}, false);
</script>

```

Рисунок 38. Код для реализации алгоритма заполнения формы

Реализация формы обратной связи считается успешной, если после заполнения полей формы, на почту администратора приходит письмо от посетителя веб-сайта организации.

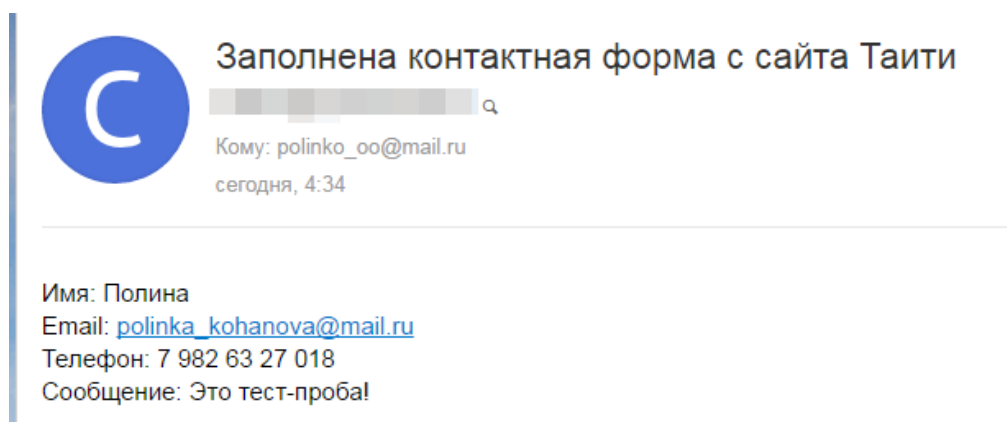


Рисунок 39. Проверка формы обратной связи

Реализация дизайна веб-сайта завершена. Имеется готовый шаблон к заполнению контентом (содержимым). Текстовый и графический материал для заполнения веб-сайта подготавливаются и предоставляются организацией. В процессе заполнения веб-сайта происходит обучение заказчика основным работам и методам управления CMS WordPress. Этап считается завершенным после предоставления заказчику исходного кода веб-сайта, модулей, баз данных и макета в формате PSD. Данные предоставляются на любом накопителе (USB-флеш карта, CD-диск и пр.).

### 2.2.3 Тестирование веб-сайта

Организации необходимо определить сотрудника, который будет работать с системой управления контентом. Обновление информации не представляет сложности, так как WordPress является интуитивно-понятной системой. Сотруднику организации детских праздников «Таити» предоставляется руководство пользователя для дальнейшего ознакомления с работой WordPress.

Чтобы понять готов ли веб-сайт, его необходимо протестировать. Тестирование веб-сайта играет важную роль в процессе разработки веб-сайта. Чем

сложнее веб-сайт, тем больше времени понадобится на проверку и отладку. Тестирование веб-сайта включает в себя следующие пункты:

- проверка работы функций веб-сайта;
- проверка гиперссылок;
- проверка работы форм (кнопки, выпадающее меню и пр.);
- проверка работы веб-сайта в разных браузерах (Chrome, Firefox, Safari, Opera);
- проверка валидности верстки.

Для проверки работы функций, гиперссылок и форм веб-сайта необходимо пользователь испытать данный веб-сайт. В процессе тестирования веб-сайта выявляются недоработки и, в дальнейшем, веб-сайт модернизируется.

Для проверки валидности верстка веб-сайта существует сайт <http://browsershots.org/>. Данная система тестирует веб-сайт в различных браузерах, которые были отмечены.

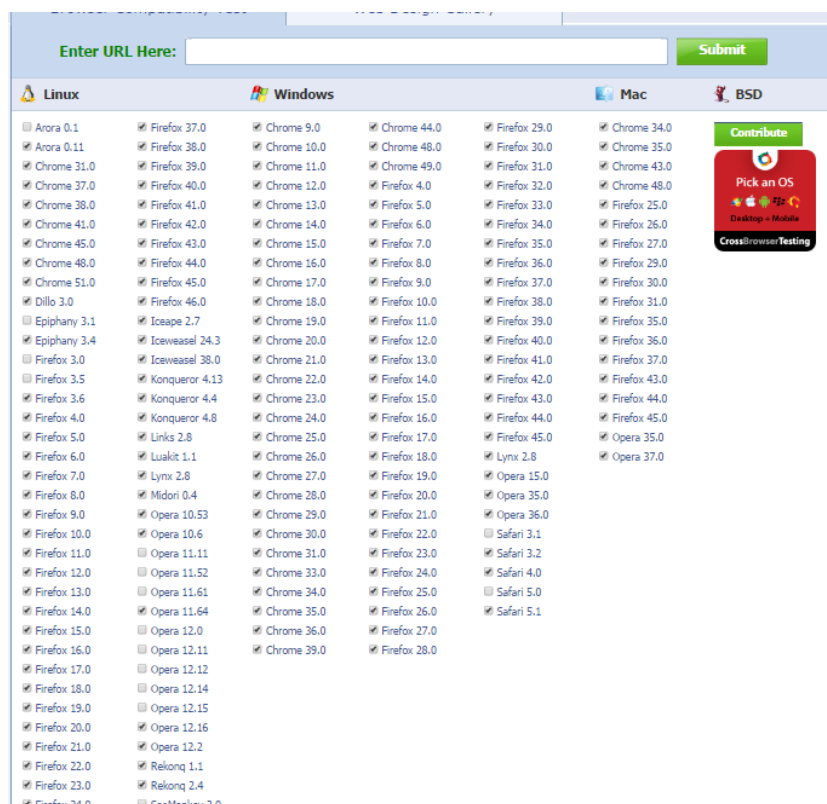


Рисунок 40. Веб-сайт <http://browsershots.org/> для проверки валидности верстки

Тестирование веб-сайта было произведено в браузерах последней версии: Chrome 39.0, Firefox 28.0, Safari 5.1, Opera 37. Так же есть возможность выбрать определенную операционную систему. Тестирование разработанного веб-сайта было на браузерах операционной системы Windows, так как данная операционная система является наиболее распространённой среди пользователей. Тестирование веб-сайта занимает время: чем больше выбрано веб-браузеров, тем больше времени уйдет на тестирование. После проведения тестирования, на сайте появляются скриншоты, сделанные с выбранных браузеров.



Рисунок 41. Проверка работы веб-сайта на Windows 7 Firefox 28.0

Валидная вёрстка – вёрстка, чья структура и код соответствуют стандартам W3C (World Wide Web Consortium). Понятие валидации есть не что иное, как проверка соответствия текущей HTML страницы со стандартами, указанными в объявлении типа документа. Валидность верстки необходима для кроссбраузерности, то есть сайты, сверстанные в соответствии со всеми стандартами, будут одинаково отображаться в большинстве браузеров. Все браузеры стараются следить за последними изменениями стандартов W3C, т.е., если страни-

ца соответствует выбранному HTML или XHTML стандарту, то с большой степенью вероятности можно сказать, что она будет отображаться одинаково во всех браузерах, которые поддерживают этот стандарт. Для проверки валидности верстки необходимо перейти на веб-сайта <https://validator.w3.org>.



Рисунок 42. Веб-сайт W3C

После ввода адреса тестируемого веб-сайта, W3C выдаст ошибки, связанные с кодом HTML-документов.

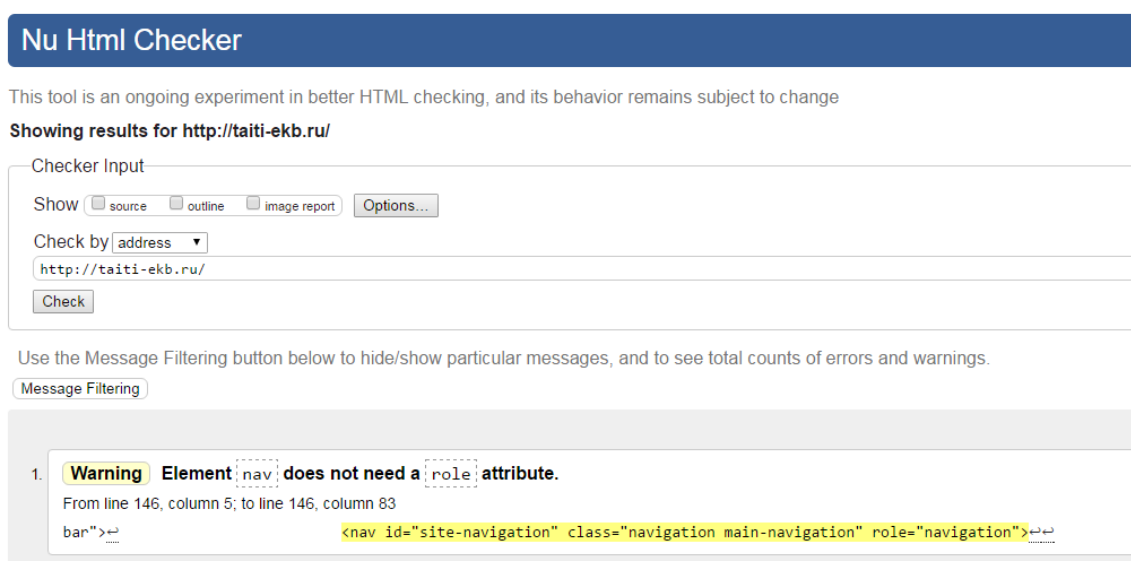


Рисунок 43. Пример работы веб-сайта W3C

Выявленные ошибки код дорабатываются, и веб-сайт снова проверяется в валидаторе.

## 2.2.4 Безопасность веб-сайта

К вопросам безопасности надо подходить серьезно. Интернет содержит множество угроз в виде вирусов и троянских программ. Все системы управления

контентом содержать уязвимости, поэтому так необходимо всегда обновлять версию WordPress. Если есть обновления, то в консоли администратора высветится сообщение об обновлении. Обновить версию можно автоматически либо вручную. Рекомендуется выполнять автоматическое обновление, так будет меньше шансов неверной установки обновления.

Еще одним немаловажным моментом по защите систему управления контентом является скрывание версии WordPress. Так как у каждой версии есть свои уязвимости и с каждым обновлением разработчики устраняют уязвимости, а злоумышленники снова их находят, то скрывание версии CMS WordPress будет еще одним шагом к защите веб-сайта. Для того, чтобы скрыть версию необходимо в файле `header.php` удалить следующую строку:

```
19 <script src="<?php echo esc_url( get_template_directory_uri() ); ?>/js/html5
20 <![endif]-->
21 <?php wp_head(); ?>
22 </head>
23
24 <body <?php body_class(); ?>>
25 <meta content="WordPress<?php bloginfo ( 'version' ); ?>" />
26 <div id="page" class="hfeed site">
27 <a class="skip-link screen-reader-text" href="#content"><?php _e( 'Skip to c
28
29 <div id="sidebar" class="sidebar">
30 <header id="masthead" class="site-header" role="banner">
31 <div class="site-branding">
32 <?php
```

Еще одним способом защиты веб-сайта является ограниченный доступ входа в администраторскую часть. Открывает окно программы FileZilla, находим в корневой папке файл `.htaccess` и указываем следующий код:

```
<Files wp-login.php>
order deny,allow
Deny from All
Allow from IP компьютера заказчика
</Files>
```

Order deny,allow - Deny проверяется все директивы — если хотя бы одна из них совпадет и нет ни одной директивы Allow, то доступ будет отвергнут. Если адрес совпадает только с директивой Allow - доступ разрешен. Если адрес не совпадает ни с одной директивой Allow или Deny, то тогда доступ тоже будет разрешен. Наиболее типичная конфигурация показана выше — запретить все, а потом разрешить доступ выбранным адресам.

Deny from – команда указывает с каких Ip-адресов запрещен доступ.

Allow from Ip – команда разрешает доступ указанному IP-адресу.

Также производится резервное копирование веб-сайта. Хостинг-провайдер предоставляет функцию резервного копирования веб-сайта. Резервное копирование происходит в панели управления веб-хостингом. Выбрав вкладку «Управление Backup» откроется окно с вкладками «Файлы», «Базы MySQL» и «Статус».

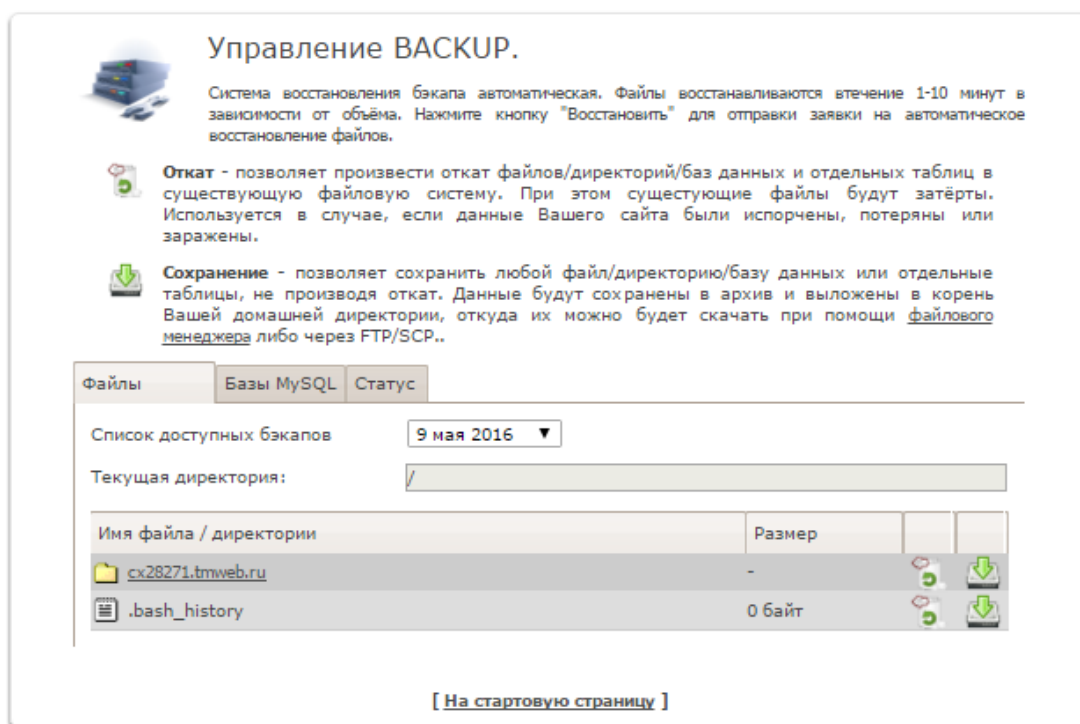


Рисунок 44. Управление BackUp

Стоит отметить, что основные методы безопасности напрасны, если компьютер, на котором размещен веб-сайт, будет подвержен атакам вирусов и троянских программ. Поэтому защита компьютера также является неотъемлемой частью по защите веб-сайта.

## 2.3 Рекомендации по использованию и настройке веб-сайта на CMS WordPress



В рамках данной работы подготовлена техническая документация по использованию и настройке веб-сайта для администратора организации детских праздников «Таити».

Администратору веб-сайта организации для начала работы с консолью необходимо иметь доступ в сеть Интернет. Переход в консоль администратора веб-сайта осуществляется с помощью перехода по ссылке <http://taiti-ekb.ru/wp-login.php>. Данные логина и пароль можно найти на веб-хостинге TimeWeb в разделе «Конструктор сайтов» - Информация «реквизиты». После ввода логина и пароля, переходим в администраторскую консоль.

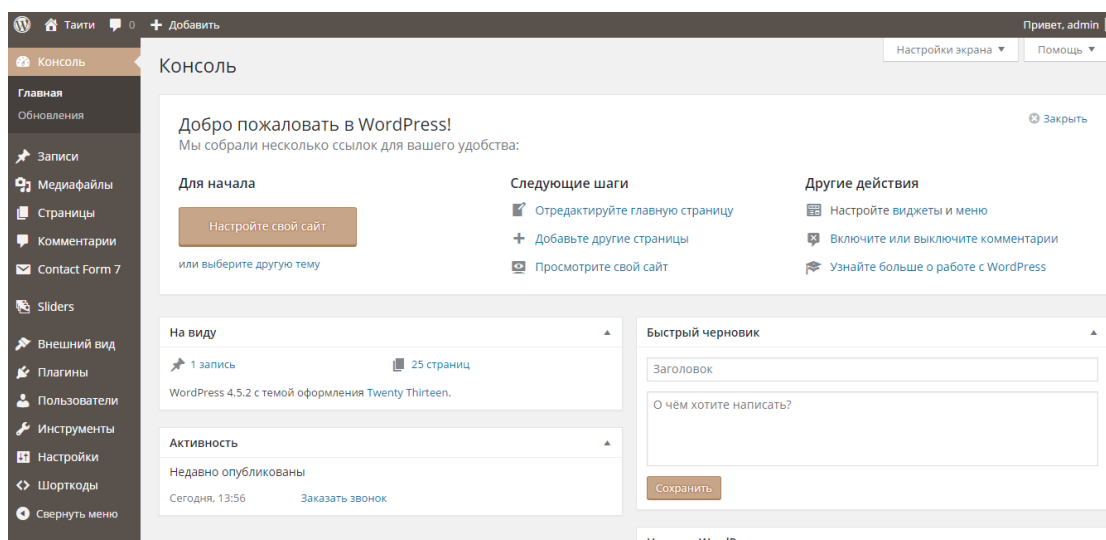


Рисунок 45. Консоль администратора

Дальше происходит работа с категориями консоли.

### Категория «Записи»

Записи — это те сообщения, которые публикуются на сайте. Записи располагаются в обратном хронологическом порядке, т.е. самая последняя запись будет расположена вверху всех сообщений (записей). Рубрики — это не что иное, как разделы блога, посвященные определенной тематике. Метка — это ключевое слово, по которому пользователь находит схожие статьи. Для добавления новой записи необходимо перейти на вкладку «Записи» - «Добавить новую». Откроется окно редактора записей.



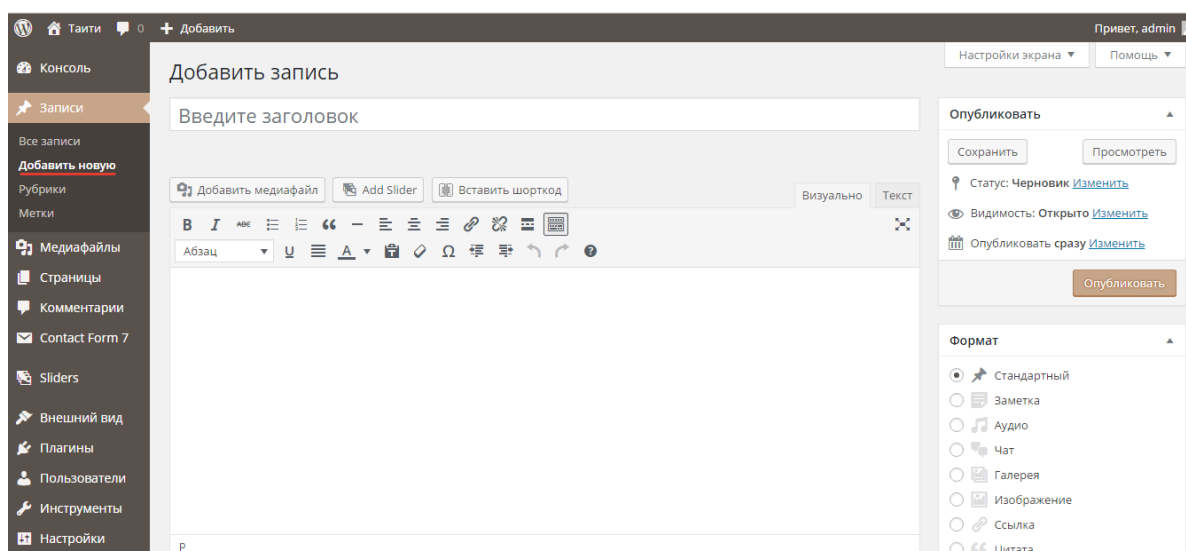


Рисунок 46. Редактор записей

Аналогично происходит добавление рубрики и метки.

### Категория «Медиафайлы»

Позволяет добавлять на сайт, файлы различных типов: изображения, аудио и видео файлы.

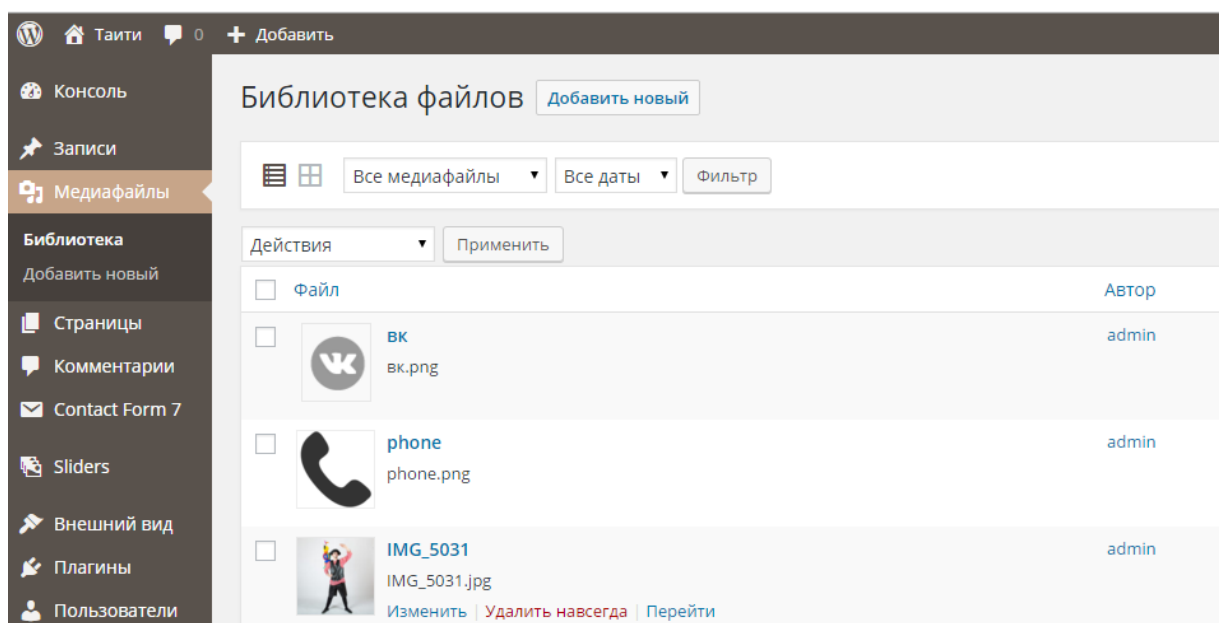


Рисунок 47. Библиотека медиафайлов

### Категория «Страницы»

Страница предназначена для такого содержимого как «Об организации», «Контактная информация» и пр. С помощью страниц можно организовать и управлять любым количеством содержимого.

Для редактирования записи или страницы достаточно перейти в (меню → страницы) или (меню → записи) и открыть нужный для контента. Возможность: Изменить, просмотреть, удалить запись или страницу.

### **Категория «Комментарии»**

Комментарии могут оставлять зарегистрированные или анонимные пользователи – это зависит от настроек сайта. Возможно: одобрить, пометить как спам, отклонить или удалить комментарий, или группу комментариев.

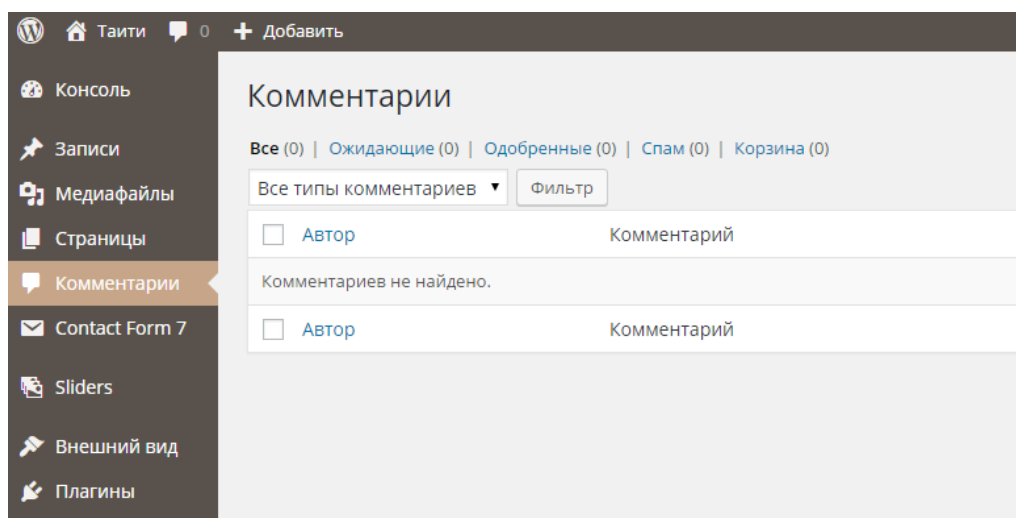


Рисунок 48. Работа с категорией «Комментарии»

### **Категория «Внешний вид»**

#### **Раздел «Меню»**

С помощью данного функционала редактируется или создается меню. Для того чтобы создать меню, требуется:

1. Ввести название меню
2. Выбрать из списка страницы или записи.
3. Поместить меню в нужном месте,
4. Выбирать родительское меню для сайта. Как правило данное меню находится в верхней части сайта.

## **Раздел «Темы»**

В WordPress предоставляет огромный выбор тем. После выбора интересующей темы, необходимо ее скачать. Распаковывать тему нужно в ftp-сервер в папку: wp-content/themes.

## **Раздел «Виджеты»**

Виджет – вспомогательные графические модули, служащие для украшения рабочего пространства, развлечения, решения отдельных рабочих задач. Для подключения на сайт зайдите в раздел «Внешний вид» → «Виджеты».

## **Раздел «Заголовок»**

Шапка сайта - это изображение, находящееся в верхней части сайта. Стандартное изображение, появляющееся после установки сайта, можно изменить, загрузив свое изображение.

## **Раздел «Фон»**

В этом разделе можно установить цвет фона нажав на кнопку: Выбрать цвет, и сохранить изменения. А также, можно загрузить изображение фона, нажав на кнопку: Обзор, и сохранить изменения.

## **Раздел «Редактор»**

Редактор позволяет редактировать исходный код выбранной темы.

## **Категория «Плагины»**

Плагины - модули для расширения базовой функциональности сайта.

Для того чтобы плагин можно было использовать, он должен быть активным. Для этого нажмите активировать под названием нужного плагина. Аналогично плагин деактивируется.

Редактор - возможность изменять исходный код плагина.

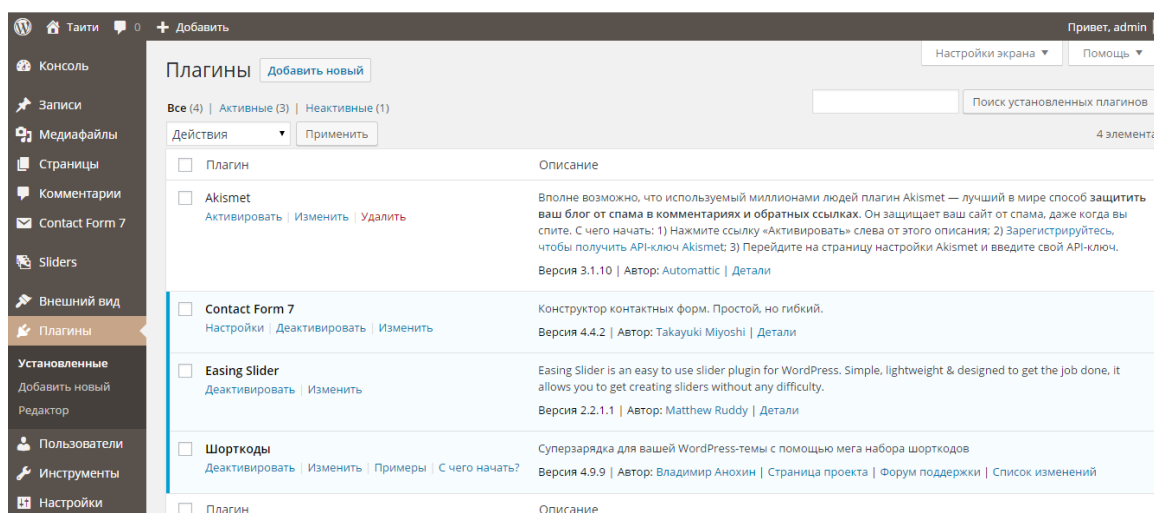


Рисунок 49. Работа с категорией «Плагин»

## Категория «Пользователи»

Зарегистрированному пользователю можно предоставить возможность добавлять статьи на сайт, дать права модератора комментариев

## Категория «Инструменты»

### Раздел «Импорт»

WordPress может импортировать записи и комментарии находящиеся на других системах. Для этого необходимо выбрать систему на которой находятся записи.

### Раздел «Экспорт»

Создаст файл с расширением WRX, которые будет содержать информацию сайта. Есть возможность выбрать материал для экспорта: все содержимое, записи, страницы, формы.

## Настройки веб-сайта

Начальные настройки веб-сайта происходят в категории «Настройки»

В разделе «Общие» указываются название и краткое описание сайта. В полях «Адрес WordPress» и «Адрес сайта» указывается URL ресурса. Так же устанавливается часовой пояс, формат даты, язык сайта.

«Адрес E-mail» - в этом поле прописан адрес электронной почты, на который администратору будут приходить служебные уведомления.

«Членство. Роль нового пользователя» в этой опции необходимо поставить галочку для регистрации пользователей на сайте.

Общие настройки

Название сайта

Краткое описание   
*Объясните в нескольких словах, о чём этот сайт.*

Адрес WordPress (URL)

Адрес сайта (URL)   
*Если вы хотите, чтобы адрес сайта отличался от директории )*

Адрес e-mail   
*Этот адрес используется в целях администрирования. Например*

Членство ☐ Любой может зарегистрироваться

Роль нового пользователя

Часовой пояс

Рисунок 50. Общие настройки веб-сайта

## Раздел «Написание»

В разделе «Написание» администратор настраивает публикацию. Обязательно указывать хотя бы один надежный сервис слежения за обновлениями в разделе «Сервисы обновления».

Настройки публикации

Основная рубрика

Основной формат записей

Публикация через e-mail

Чтобы публиковать записи в WordPress при помощи e-mail, вам необходимо создать скрытый e-mail с POP3-доступом. Лк произвольных имени, которые вы можете использовать: 'zXAIIiBM', hPu0AEUg, 9UJAR5MP.

Почтовый сервер  Порт

Логин

Пароль

Рубрика по умолчанию для публикации по почте

Сервисы обновления

Когда вы публикуете новую статью, WordPress автоматически уведомляет следующие сайты служб слежения за обновлен несколько, пишите каждый в новой строке.

Рисунок 51. Настройки публикации

## Раздел «Чтение»

Раздел «Чтение» позволяет настроить информацию, которая будет выводиться на главную страницу сайта: последние записи или статические страницы.

## Настройки чтения

На главной странице отображать

☐ Ваши последние записи

☒ Статическую страницу (выберите ниже)

Главная страница: #163 (без названия) ▼

Страница записей: — Выбрать — ▼

На страницах блога отображать не более 10 записей

В RSS-лентах отображать последние 10 элементов

Для каждой статьи в RSS-ленте отображать

☒ Полный текст

☐ Анонс

Видимость для поисковых систем

☐ Попросить поисковые системы не индексировать сайт

*Будет ли учитываться этот запрос — зависит от поисковых систем.*

Рисунок 52. Настройки чтения

## Раздел «Обсуждение»

В разделе «Обсуждение» есть возможность настройки комментирования, чтобы сайт не заполнился спамом.

## Настройки обсуждения

Помощь ▼

Настройки для статьи по умолчанию

☒ Пытаться оповестить блоги, упоминаемые в статье

☒ Разрешить оповещения с других блогов (уведомления и обратные ссылки) на новые статьи

☐ Разрешить оставлять комментарии на новые статьи

*(Эти настройки могут быть изменены для каждой отдельной статьи.)*

Другие настройки комментариев

☒ Автор комментария должен указать имя и e-mail

☐ Пользователи должны быть зарегистрированы и авторизованы для комментирования.

☐ Автоматически закрывать обсуждение статей старше 14 дней

☒ Разрешить древовидные (вложенные) комментарии глубиной 5 уровней

☐ Разбивать комментарии верхнего уровня на страницы по 50 штук, по умолчанию отображается последняя страница

Сверху каждой страницы должны располагаться ранние комментарии

Отправить мне письмо, когда

☐ Кто-нибудь оставил комментарий

☒ Комментарий ожидает проверки

Перед появлением комментария

☐ Комментарий должен быть одобрен вручную

☒ Автор должен иметь ранее одобренные комментарии

Рисунок 53. Настройки обсуждения

## Раздел «Медиафайлы»

Раздел «Медиафайлы» отвечает за размещение и отображение изображений и других медиафайлов на сайте при их загрузке на сервер из панели управления.

Настройки медиафайлов

Размеры изображений

Указанные ниже числа определяют максимальные размеры изображения в пикселях при добавлении его в библиотеку файлов.

Размер миниатюры      Ширина  Высота   
☒ Обрезать миниатюру точно по размерам (обычно сохраняются пропорции миниатюр)

Средний размер      Макс. ширина  Макс. высота

Крупный размер      Макс. ширина  Макс. высота

Загрузка файлов

☒ Помещать загруженные мной файлы в папки по месяцу и году

Рисунок 54. Настройки медиафайлов

## Раздел «Постоянные ссылки»

В разделе «Постоянных ссылок» можно выбрать один из четырех готовых вариантов отображения URL-адресов страниц сайта. Кроме этого в пятом варианте вы можете создать свой собственный шаблон.



## Настройки постоянных ссылок

WordPress даёт возможность указать для постоянных ссылок и архивов собственную структуру. Это может придать вашим ссылкам эстетичный вид, а для начала — несколько примеров.

### Общие настройки

<input type="radio"/> Простые	<code>http://taiti-ekb.ru/?p=123</code>
<input checked="" type="radio"/> День и название	<code>http://taiti-ekb.ru/2016/05/11/sample-post/</code>
<input type="radio"/> Месяц и название	<code>http://taiti-ekb.ru/2016/05/sample-post/</code>
<input type="radio"/> Цифры	<code>http://taiti-ekb.ru/archives/123</code>
<input type="radio"/> Название записи	<code>http://taiti-ekb.ru/sample-post/</code>
<input type="radio"/> Произвольно	<code>http://taiti-ekb.ru /%year%/%monthnum%/%day%/%postname%/</code>

### Дополнительно

Если хотите, можно добавить произвольные префиксы для URL-адресов рубрик и меток. Например, `/topics/` сделает ссылки на рубрики значения по умолчанию.

Префикс для рубрик

Префикс для меток

Рисунок 55. Настройка постоянных ссылок

Написав инструкции по использованию и настройке веб-сайта, осуществляется обучение заказчика или сотрудника организации детских праздников «Таити». Исполнитель показывает работу в администраторской консоли, объясняя смысл своих действий. После обучения происходит перенос всех архивов, модулей на накопитель и передается заказчику. Также передается вся информация о веб-сайте (логин, пароль от веб-хостинга, почтовый ящик администратора). После передачи всей информации работа считается оконченной.

## **Заключение**

В современном мире веб-технологии становятся уникальным средством предоставления информации. Использование веб-сайта для организации рекламной деятельности значительно влияет на расширение целевой аудитории. Цель разработки – создание веб-сайта для организации детских праздников «Таити» – достигнута.

Для достижения поставленной цели были выполнены следующие задачи:

1. Произведен сравнительный анализ CMS и дано обоснование выбора CMS.
2. Дано обоснование выбора программных средств и освоен процесс разработки веб-сайта.
3. Произведена разработка дизайн-макета веб-сайта.
4. Произведена установка и настройка системы управления содержанием.
5. Произведена установка и настройка компонентов, модулей и плагинов, необходимых для функциональности веб-сайта.
6. Произведена вёрстка макета веб-сайта.
7. Проведено обучение сотрудников Заказчика редактированию веб-сайта.

Поставленные задачи выполнены, основная цель исследования – достигнута. Произведенная работа носит законченный характер.

## Список информационных источников

1. Развитие интернета в регионах России // yandex.ru URL: [https://yandex.ru/company/researches/2015/ya\\_internet\\_regions\\_2015/#auditorijai-interneta](https://yandex.ru/company/researches/2015/ya_internet_regions_2015/#auditorijai-interneta) (дата обращения: 14.01.2016).
2. Документация для разработчика «WordPress Codex». URL: <http://codex.wordpress.org/> (дата обращения: 20.01.2016).
3. Установка WordPress // wp-kama.ru URL: <http://wp-kama.ru/handbook/wordpress/ustanovka-wordpress> (дата обращения: 25.01.2016).
4. Wordpress начинающим // wordpressinside.ru URL: <http://wordpressinside.ru/beginners/> (дата обращения: 25.01.2016).
5. Рейтинг CMS по версии iTrack // itrack.ru URL: <http://www.itrack.ru/research/cmsrate/> (дата обращения: 25.01.2016).
6. Создание шаблона для WordPress // habrahabr.ru URL: <https://habrahabr.ru/post/239653/> (дата обращения: 26.02.2016).
7. Создание темы wordpress // unix-lab.org URL: <http://www.unix-lab.org/posts/wp-theme/> (дата обращения: 25.02.2016).
8. 20 советов по безопасности WordPress // blogproblog.com URL: <https://blogproblog.com/wordpress-security/> (дата обращения: 05.03.2016).
9. Верстка шаблона для WordPress. Часть 1 // font-family.ru URL: <http://font-family.ru/verstka-shablona-dlya-wordpress-chast-1/> (дата обращения: 30.03.2016).
10. Верстка шаблона для WordPress. Часть 2 // font-family.ru URL: <http://font-family.ru/verstka-shablona-dlya-wordpress-chast-2/> (дата обращения: 30.03.2016).
11. Верстка шаблона для WordPress. Часть 3 // font-family.ru URL: <http://font-family.ru/verstka-shablona-dlya-wordpress-chast-3/> (дата обращения: 30.03.2016).

12. Вёрстка сайтов // htmlbook.ru URL: <http://htmlbook.ru/samlayout> (дата обращения: 31.03.2016).
13. Вёрстка сайта - шпаргалка // internet-technologies.ru URL: [http://www.internet-technologies.ru/articles/article\\_1856.html](http://www.internet-technologies.ru/articles/article_1856.html) (дата обращения: 03.04.2016).
14. Правильное тестирование веб-сайта, или как обеспечить себе спокойный сон // fresh-design.com.ua URL: <http://fresh-design.com.ua/blog/technology/website-testing> (дата обращения: 25.04.2016).
15. Этапы тестирования веб-сайтов // art-lemon.com URL: <https://art-lemon.com/site-test> (дата обращения: 25.04.2016).
16. Тестирование и сравнение нескольких плагинов для кэширования WordPress // wpcafe.org URL: <https://wpcafe.org/plugins/wordpress-cache-plugin-benchmarks/> (дата обращения: 25.04.2016).
17. Плагины для wordpress // my-wordpress.ru URL: <http://my-wordpress.ru/plaginy-wordpress> (дата обращения: 27.04.2016).
18. Теги WordPress // wp-info.ru URL: <http://www.wp-info.ru/tegi-wordpress/> (дата обращения: 28.04.2016).
19. Основы HTML // htmlbook.ru URL: <http://htmlbook.ru/content/osnovy-html> (дата обращения: 29.04.2016).
20. Основы CSS // htmlbook.ru URL: <http://htmlbook.ru/content/osnovy-css> (дата обращения: 29.04.2016).
21. Структура шаблона страницы темы WordPress // mblogm.ru URL: <http://mblogm.ru/blog/struktura-shablona-stranitsy-temy-wordpress/> (дата обращения: 05.05.2016).
22. Уроки по HTML, CSS // life-prog.ru URL: [http://life-prog.ru/view\\_cat.php?cat=7](http://life-prog.ru/view_cat.php?cat=7) (дата обращения: 06.05.2016).
23. Основы основ PHP: обзор для начинающих // codeharmony.ru URL: <http://www.codeharmony.ru/materials/30> (дата обращения: 06.05.2016).
24. Брайн Хоган HTML5 и CSS3. Веб-разработка по стандартам нового поколения. 2 изд. СПб.: Питер, 2012. 270 с.

25. Прохоренок Н.С. HTML, JavaScript, PHP и MySQL. Джентльменский набор Web-мастера. 3 изд. СПб.: БХВ-Петербург, 2010. 900 с.
26. Фрейен Бен HTML5 и CSS3. Разработка сайтов для любых браузеров и устройств. СПб.: Питер, 2014. 304 с.
27. Робин Никсон HTML 5, CSS 3 и Web 2.0. Разработка современных Web-сайтов. 3 изд. СПб.: Питер, 2015. 688 с.
28. Дженнифер Нидерст Роббинс HTML5, CSS3 и JavaScript. Исчерпывающее руководство. СПб.: Эксмо, 2014. 528 с.
29. Дженнифер Нидерст Роббинс Web-дизайн. Справочник. М.: КУДИЦ-Пресс, 2015. 354 с.
30. Якоб Нильсен, Хоа Лоранжер Web-дизайн: удобство использования Web-сайтов. СПб.: Питер, 2014. 368 с.